The Dissertation Committee for Subhashini Venugopalan certifies that this is the approved version of the following dissertation:

Natural-Language Video Description with Deep Recurrent Neural Networks

Committee:

Raymond J. Mooney, Supervisor

Kristen Grauman

Peter Stone

Kate Saenko

Trevor Darrell

Natural-Language Video Description with Deep Recurrent Neural Networks

by

Subhashini Venugopalan,

Dissertation

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
August 2017

To my family

Acknowledgments

Over the last few years, I have had the excellent fortune of interacting with some wonderful people, remarkable mentors and friends who have offered me support and guidance and helped shape an important part of my life.

I am very grateful to my advisor Raymond Mooney, for his guidance and generosity. Ray’s foresight, curiosity, passion, and vision as a scientist and researcher have been very inspiring. Over our many meetings, his vast knowledge, and his uncanny ability to look at ambitious projects and point to the appropriate foothold have helped me develop and grow as a researcher. I am grateful to him for giving me the freedom to pursue my ideas and providing helpful advice on a regular basis. Ray’s dedication to the ML group at UT has been extraordinary, and I have also benefited significantly from his collaborative attitude to research, in particular I’m indebted to him for introducing me to Kate and Trevor.

I have been extremely fortunate to also have the opportunity to work closely with Kate Saenko and Trevor Darrell who have been exceptional co-supervisors, and for their insightful advice on much of the research presented in this thesis. I would like to thank Kate for her continuous encouragement and guidance, and Trevor for providing unique perspectives, and excellent feedback and advice on all aspects of research. I am also grateful to Trevor for introducing me to many of my collaborators and for providing me with additional computational resources. I owe both Kate and Trevor a lot of gratitude for their generous time and mentorship these last few years and for putting up with me through several paper deadlines.

I would also like to thank Peter Stone and Kristen Grauman for their willingness to be on my committee and for providing me thoughtful feedback on this thesis. Sections in this thesis have been inspired by some of Kristen’s work and I’m grateful to her for the discussions we have had and her time. I would also like to thank Dana Ballard, Joydeep Ghosh, Pradeep Ravikumar, Katrin Erk, Varun Rai, William Press, Adam Klivans, Lorenzo Alvisi, Vitaly Shmatikov, and Brent Waters all of whom I have had the pleasure of interacting with during my time at UT and who have all in some ways helped shape my approach to research.

It has been great to have worked with some wonderful collaborators along the way. My thanks to Jeff Donahue, Lisa Hendricks, Ronghang Hu, Sergio Guadar-

rama, Niveda Krishnamoorthy, Girish Malkarnenkar, Dong Huk Park, Raymond Ptucha, Vasili Ramanishka, Marcus Rohrbach, Sagan Sah, Jesse Thomason, and Huijuan Xu. Special thanks also to helpreq and the UTCS staff and sys-admins.

I would also like to thank my fantastic podmates Matthew Hausknecht, Karl Pichotta, and Wesley Tansey and pseudo-podmate Stephen Roller for stimulating discussions on research, philosophy, and life, and the sheer number of ideas that our conversations have generated. My thanks also to many wonderful colleagues: Amelia Harrison, Dinesh Jayaraman, Elad Liebman, Islam Beltagy, Yinon Bentor, Piyush Khandelwal, Suyog Jain, Venkata Koppula, Matteo Pontecorvi, Katie Genter, Nazneen Rajani, Aishwarya Padmakumar, Akansha Saran, Surbhi Goel, Josiah Hanna, and Ashish Bora for making grad-life at UT very memorable.

During my PhD, I had the distinct pleasure of spending multiple summers as an intern at Google meeting amazing people and learning many new things. I would like to particularly thank my hosts Varun Gulshan, Martin Stumpe and Andrea Frome, who have helped hone my research skills, and the many mentors: Kevin Murphy, Vincent Vanhoucke, Jeff Dean, Oriol Vinyals, Samy Bengio, Geoff Hinton, George Toderici, Dale Webster, Arun Narayanaswamy, Marc Berndl, and Philip Nelson for the wonderful learning experience.

My PhD endeavor owes itself to the support of excellent mentors during my masters and undergraduate days. In particular, I want to thank Prof. C. Pandu Rangan and Prof. Ananthanarayana VS for introducing me to research and for always believing in me, and Ramarathnam 'Venkie' Venkatesan for his guidance and encouragement at critical moments.

Thanks also to the many friends who have made my stay in Austin pleasant and enjoyable: Shruthi Vishwanath, Ashay Rane, Hiran Thyagarajan, Madhura Parikh, Meghana Nasre, Rupesh Nasre, Sharayu Moharir, Kumar Appaiah, Akarsh Simha, Akanksha Jain, and Siva Ramakrishnan. Special thanks to the wonderful folks at the Austin chapters of the Sai center, Naadhasudha (Vishaal Sapuram), and Samskrita Bharathi for making Austin a home away from home.

Finally, thanks to my family for their love, support, and just everything. None of this would have been possible without their sacrifices and encouragement, and for that I'm always indebted.

Natural-Language Video Description with Deep Recurrent Neural Networks

by

Subhashini Venugopalan, Ph.D.
The University of Texas at Austin, 2017
Supervisor: Raymond J. Mooney

For most people, watching a brief video and describing what happened (in words) is an easy task. For machines, extracting meaning from video pixels and generating a sentence description is a very complex problem. The goal of this thesis is to develop models that can automatically generate natural language descriptions for events in videos. It presents several approaches to automatic video description by building on recent advances in "deep" machine learning. The techniques presented in this thesis view the task of video description akin to machine translation, treating the video domain as a source "language" and uses deep neural net architectures to "translate" videos to text. Specifically, I develop video captioning techniques using a unified deep neural network with both convolutional and recurrent structure, modeling the temporal elements in videos and language with deep recurrent neural networks. In my initial approach, I adapt a model that can learn from paired 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. To further improve grammaticality and descriptive quality, I also propose methods to integrate linguistic knowledge from plain text corpora. Additionally, I show that such linguistic knowledge can help describe novel objects unseen in paired image/video-caption data. Finally, moving beyond short video clips, I present methods to process longer multi-activity videos, specifically to jointly segment and describe coherent event sequences in full-length movies.

Table of Contents

List of Tables

List of Figures

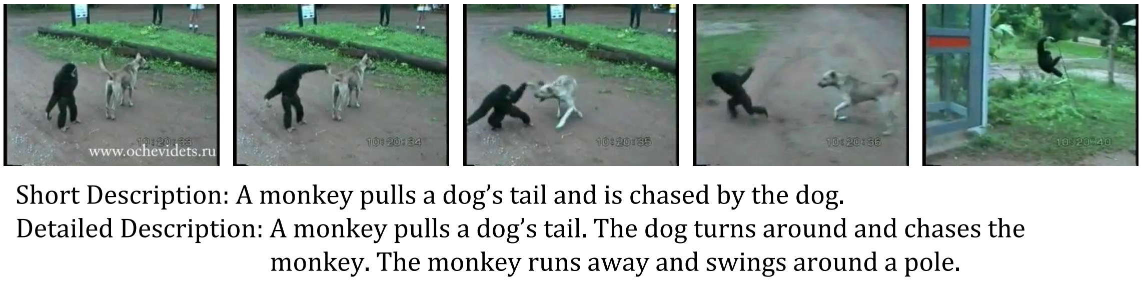
Chapter 1

Introduction

Videos are a rich and complex source of information, and they constitute the largest chunk of the content on the internet. For most humans, understanding multimedia content is easy, and in many cases images and videos are a preferred means of augmenting and enhancing human interaction and communication. Given a video, humans can discern a great deal from this rich information source and can interpret and describe the content to varying degrees of detail e.g. as a succinct summary, or even as a detailed sequence of events (Figure 1.1). For computers however, interpreting content from image and video pixels is very challenging. The goal of research in language and vision is to develop intelligent systems that can autonomously analyze and understand this complex visual data as well as interact and express itself in natural language.

This dissertation looks fundamentally at the problem of describing content in videos. The ability to automatically describe videos in natural language 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 among others. In this chapter I will first outline some of the challenges in video understanding and description, and then highlight contributions of my research.

Challenges. The core of video description or captioning, consists of three main research components, object recognition, activity recognition, and *surface realization* (or sentence generation). Early work on natural language description of visual data focused primarily on static images (Yao *et al.*, 2010, Farhadi *et al.*, 2010, Kulkarni *et al.*, 2011, Li *et al.*, 2011, Kuznetsova *et al.*, 2012, Yang *et al.*, 2011). These relied on several algorithms and techniques for recognizing objects in images, and simple template based approaches for generating a sentence. There have been a few research works that have extended these methods to video description (Kojima *et al.*, 2002, Lee *et al.*, 2008, Khan and Gotoh, 2012, Barbu *et al.*, 2012, Ding *et al.*, 2012, Das *et al.*, 2013b;a, Rohrbach *et al.*, 2013, Yu and Siskind, 2013) but only within narrow domains (e.g., cooking), which contain limited vocabularies of objects and activities. These works relied heavily on the specific domain in order to



Short Description: A monkey pulls a dog's tail and is chased by the dog.
Detailed Description: A monkey pulls a dog's tail. The dog turns around and chases the monkey. The monkey runs away and swings around a pole.

Figure 1.1: Describing activities depicted in videos require integration of both visual and linguistic capabilities. For most humans this is easy. This example presents frames from a YouTube video clip and human generated descriptions of different granularity.

build good object and activity classifiers, hence they were quite difficult to generalize to videos 'in-the-wild'.

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. Early work by Krishnamoorthy *et al.* (2013) and Guadarrama *et al.* (2013a) achieved promising results on the task of generating natural language descriptions of short open-domain video clips by relying primarily on good object recognition techniques and template based sentence generation. However, the task of 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 still remained a challenge. 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.

Progress. These last few years, deep machine learning approaches have achieved remarkable success on object recognition tasks and sequence modeling in natural language, sparking renewed interest in image and video captioning. In particular, deep convolutional neural network approaches (Krizhevsky *et al.*, 2012) achieved ground breaking results on object recognition tasks in large image datasets (Deng *et al.*, 2009b). Closely following this, deep recurrent neural network approaches achieved resounding success on sequence modeling tasks in natural language processing, machine translation in particular (Bahdanau *et al.*, 2014, Sutskever *et al.*, 2014) and speech recognition (Graves *et al.*, 2013). These in turn lead to a marked increase in work on natural-language image description. Notably, several deep neural network based methods (Donahue *et al.*, 2015, Chen and Zitnick, 2015, Karpathy and Fei-Fei, 2015, Kiros *et al.*, 2015, Kuznetsova *et al.*, 2014, Mao *et al.*, 2014, Vinyals *et al.*, 2015, Fang *et al.*, 2015) achieved breakthrough results on the task of describing images with a single sentence. In contrast, video description has seen far less attention and deep neural network approaches to image captioning do not address the problem of modeling a sequence of visual inputs and describing them in full sentences.

Contributions. This thesis presents some of the first fully deep models for video captioning. We leverage transformative advances in "deep" machine learning combining them with the latest techniques in computer vision and natural language processing (NLP) to develop improved and scalable methods for natural-language video description.

In previous work (Thomason *et al.*, 2014), we focused on addressing the issue of describing open-domain videos with large vocabularies by integrating linguistic knowledge with visual recognition. As seen from the example in Figure 1.1, describing activities depicted in video requires integrating both visual and linguistic capabilities. 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 subject-verb-object-scene (SVOP) tuple that can be used to describe a short video clip. As mentioned previously, scaling such models to describe salient elements is difficult.

Inspired by deep image captioning models (Donahue *et al.*, 2015, Vinyals *et al.*, 2015), I present the first fully deep model for video description (Venugopalan *et al.*, 2015b). We use deep recurrent neural networks based on Long Short Term Memory (LSTM, Hochreiter and Schmidhuber (1997)) 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 Venugopalan *et al.* (2015a), 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. We also show the generality of the approach by describing short clips from movies.

I then develop extensions to incorporate prior linguistic knowledge into deep video and image captioning models. I propose multiple techniques to integrate knowledge from plain text corpora to improve both grammaticality and descriptive quality in video captioning (Venugopalan *et al.*, 2016). In addition, I show that such methods of integrating linguistic knowledge can be particularly helpful to describe novel objects unseen in paired image-caption training data (Venugopalan *et al.*, 2017). 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, in particular to generate descriptions of longer clips from movies.

Organization

The remainder of this work is organized in 9 chapters: Chapter 2 presents background about initial works in video description and an introduction to deep recurrent neural networks; Chapters 3 and 4 present the first recurrent neural network based approaches to video captioning; Chapter 5 presents extensions to incorporate statistical language models to improve descriptive quality of videos, Chapter 6 shows how such linguistic knowledge can also be used to describe novel objects, particularly in images. Chapter 7 then proceeds to discuss steps to address multi-activity videos looking more closely at the task of generating descriptions of movies. Finally, Chapter 8 places this dissertation in context discussing contemporaneous and related works, and Chapter 9 looks at some of the future directions.

Chapter 2

Background

In this chapter, I briefly 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 describe how deep recurrent neural networks (RNNs) are used to model sequences. Specifically I will examine a specific variant of RNNs termed Long Short-Term Memory (LSTM) RNNs which have been popular in machine translation and image-captioning research.

2.1 Language and Vision

Both natural language processing (NLP) and computer vision (CV) have made great strides in recent years (Jurafsky and Martin, 2009, Forsyth and Ponce, 2011), 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 a couple of years back, there was a small but growing body of work at the intersection of NLP and CV on topics like connecting words to pictures (Barnard *et al.*, 2003, Berg *et al.*, 2004, Deng *et al.*, 2009a), describing images in natural language (NL) (Farhadi *et al.*, 2010, Kulkarni *et al.*, 2011, Li *et al.*, 2011), and comprehending NL instructions in terms of robot perception and action (Matuszek *et al.*, 2010, Tellex *et al.*, 2011, Kruijff *et al.*, 2007, Guadarrama *et al.*, 2013b).

The last couple of years saw a dramatic increase in image captioning and retrieval works (Donahue *et al.*, 2015, Vinyals *et al.*, 2015, Karpathy and Fei-Fei, 2015, Kiros *et al.*, 2015, Fang *et al.*, 2015, Mao *et al.*, 2014) owing to the release of large image captioning datasets MSCOCO (Lin *et al.*, 2014) and Flickr30k (Hodosh *et al.*, 2014). Shortly following this, there have also been datasets for image question answering (Malinowski and Fritz, 2014, Antol *et al.*, 2015, Ren *et al.*, 2015a, Yu *et al.*, 2015). In comparison, progress in video description has been slower.

2.2 Early Progress in Video Description

Video description, particularly for large-vocabularies and activities 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 (Khan and Gotoh, 2012, Lee *et al.*, 2008, Kojima *et al.*, 2002, Ding *et al.*, 2012, Krishnamoorthy *et al.*, 2013, Das *et al.*, 2013b;a). Work on large-vocabulary description has focused mostly on nouns/adjectives, specifically, early work on videos considered tagging videos with metadata (Aradhye *et al.*, 2009) and clustering captions and videos (Huang *et al.*, 2013, Over *et al.*, 2012, Wei *et al.*, 2010) for retrieval tasks.

Early work on video description used hand-crafted templates, grammars and rules, and worked in fairly constrained domains. For example, Barbu *et al.* (2012) and Yu and Siskind (2013) produce sentential descriptions for short videos but only recognize 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, Guadarrama *et al.* (2013a) and Krishnamoorthy *et al.* (2013) use a two stage pipeline that first identifies the semantic content (subject, verb, object) and then generates a sentence based on a template. In Krishnamoorthy *et al.* (2013) 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. Krishnamoorthy *et al.* (2013) used a limited set of videos containing a small set of 20 entities. Guadarrama *et al.* (2013a) was the first to describe "in-the-wild" videos with large vocabularies. They showed the advantage of using linguistic knowledge, but only for the case of "zero shot activity recognition", in which the appropriate verb for describing the activity was never seen during training.

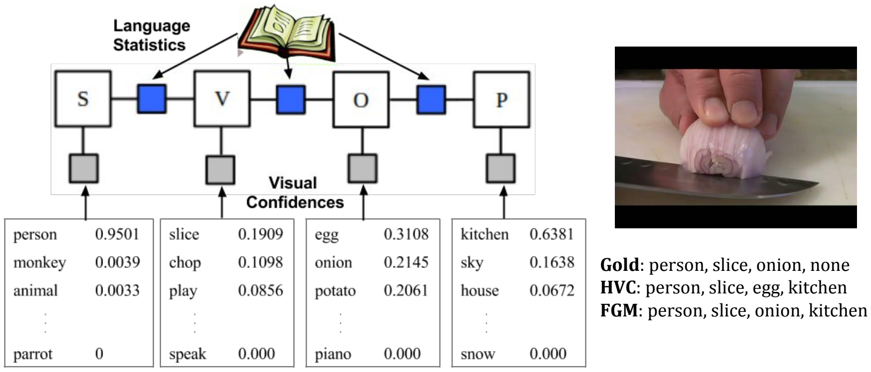


Figure 2.1: The factor graph model for estimating the most likely subject-verb-object-place (SVOP) tuple by combining confidences from visual detectors and statistics from language. (Right) The factor graph model (denoted FGM) correctly predicts "person, slice, onion, kitchen" whereas the vision system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

2.3 Integrating Language and Vision using Factor Graphs

In previous work (Thomason *et al.*, 2014), 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 Guadarrama *et al.* (2013a) to first build object and action classifiers. For detecting objects, we use ObjectBank (Li *et al.*, 2010) and the LLC-10k classifiers of Deng *et al.* (2012) trained on ImageNet 2011 with 10k object categories. Our action classifiers used features from Wang *et al.* (2011) (dense trajectories), and the features for scene recognition were based on Xiao *et al.* (2010). We trained non-linear Support Vector Machines (SVMs) (Chang and Lin, 2011) 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 (De Marneffe *et al.*, 2006) 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 Deep Neural Networks

A major short-coming of all the early image and video description approaches was that, in order to scale them one needed to pre-select and build classifiers for a range of objects, actions, and scenes, and also devise methods to identify salient objects worth describing. Advances in deep machine learning, in particular deep convolutional neural networks (CNNs) (Krizhevsky *et al.*, 2012), and deep recurrent neural networks (RNNs) (Bahdanau *et al.*, 2014, Sutskever *et al.*, 2014) helped overcome both these challenges. Both deep CNNs and RNNs form the foundation of the work presented in this thesis, hence we present a very brief overview of both CNNs and RNNs from the perspective of how they are used in this thesis.

The framework of our video description models are based on deep neural networks, in particular, CNNs for modeling visual data and Long Short-Term Memory (LSTM) RNN units for modeling language. While CNNs have shown phenomenal success on object recognition tasks (Krizhevsky *et al.*, 2012, Simonyan and Zisserman, 2014b, Szegedy *et al.*, 2015), LSTMs have recently shown superior performance on tasks such as speech recognition (Graves and Jaitly, 2014), machine translation (Sutskever *et al.*, 2014, Cho *et al.*, 2014) and the more related task of generating sentence descriptions of images (Donahue *et al.*, 2015, Vinyals *et al.*, 2015). This section aims to provide a brief overview of convolutional neural networks and recurrent neural networks, specifically Long Short-Term Memory (LSTMs) networks with focus on sequence modeling.

2.4.1 Convolutional Neural Networks

Convolutional neural networks (LeCun *et al.*, 1995) are specifically designed architectures of neural networks to efficiently process and handle data with some fixed spatial topology such as images, or fixed-length sequences of words. The goal of a CNN is to typically learn a spatially invariant representation of the input, particularly useful for classification tasks. This is achieved by neural net architectures consisting of *convolutional* and *pooling* layers described below.

Convolutional layer. The goal of a convolutional layer is to automatically learn weights of convolutional filters that can be helpful in identifying local features like detection of edges, and curvatures. The input to a CNN is typically a multi-dimensional array (or a tensor). In the case of color images, it would be a 256x256x3 tensor. And a filter is a set of weights or parameters e.g.. it could be of dimension say 5x5x3. In order for the CNN to learn local features, the filter is (mathematically) convolved over the entire input tensor to obtain an *activation map*. Intuitively, the filter weights help the model "learn" or "look for" local features that (would show up in the *activation map*, and) can help in downstream tasks. Each convolutional layer, learns not just a single filter, but a collection of filters. The weights and parameters of these filters are learned through back-propagation.

Pooling layer. Pooling layers are often used in CNNs to decrease the size of the representation learned by the convolutional layer by applying a fixed downsampling transformation. The downsampling is done spatially at the cost of losing some local spatial information, but overall this helps the network learn a good condensed representation of the input.

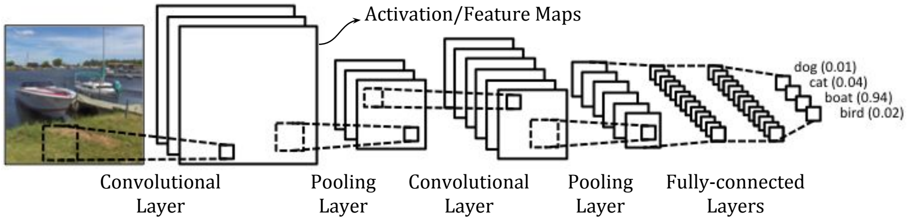


Figure 2.2: A simple convolutional neural network architecture 2 convolutional layers, 2 pooling layers, and 2 fully-connected layers.

Most CNN architectures have multiple convolutional layers followed by a pooling layer, and this structure could be repeated multiple times to create stacks of convolutional layers, and pooling layers, finally ending with fully connected layers and a final classification objective as shown Figure 2.2. The activations just before the classification layer could serve as a condensed representation of the image. All the parameters of the network are trained using back-propagation.

2.4.2 Recurrent Neural Networks

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

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

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

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.

2.4.3 Long Short-Term Memory RNNs

RNNs can learn to map sequences for which the alignment between the inputs and outputs is known ahead of time (Sutskever *et al.*, 2014) however it was unclear if they could be applied to problems where the inputs (x_i) and outputs (z_i) are of varying lengths. This problem was 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 (Cho *et al.*, 2014). 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 (Hochreiter *et al.*, 2001). However, LSTMs (Hochreiter and Schmidhuber, 1997), which incorporate explicitly controllable memory units, are known to be able to learn long-range temporal dependencies. In our work we use the LSTM unit in Figure 2.3, described in Zaremba and Sutskever (2014), and Donahue *et al.* (2015).

At the core of the LSTM model is a memory cell c which encodes each input, creating a condensed representation of the sequence of 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 incoming value from the gate (if the layer evaluates to 1) or

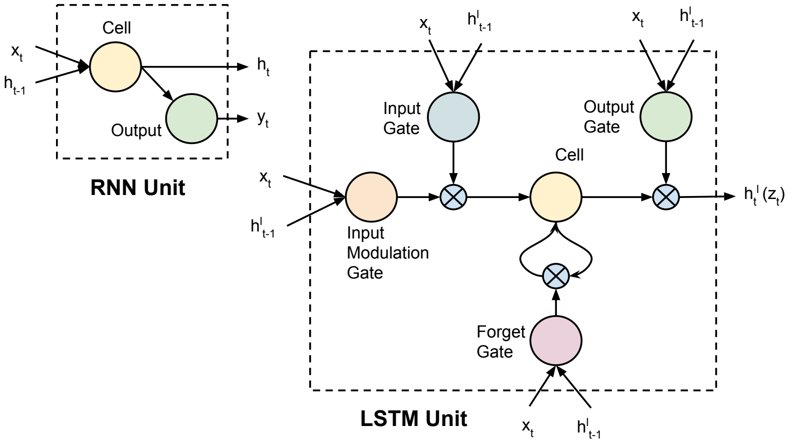


Figure 2.3: The RNN and LSTM units (replicated from Donahue *et al.* (2015)). The memory cell is at the core of the LSTM unit and it is modulated by the input, output and forget gates controlling how much knowledge is transferred at each time step.

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:

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

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

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

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

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

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.

Thus, the gates in the LSTM allow it represent a sequence by learning long-term dependencies. Hence, LSTM RNNs can "encode" a sequence of inputs to a vector, and also "decode" the vector to produce a sequence of outputs. The next chapter presents a model that employs the LSTM RNN to “decode" a visual feature vector representing the video to generate textual output. Following that, Chapter 4 describes models where LSTM networks are used to both encode a sequence of video frames to generate a visual feature vector, and decode that vector to generate a description.

Chapter 3

Video Description with Deep Recurrent Neural Networks

This chapter presents the first model that uses deep recurrent neural networks based on Long Short Term Memory (LSTM, Hochreiter and Schmidhuber (1997)) to generate video descriptions[1]. Unlike, early image and video description models seen in Chapter 2, our LSTM based model learns what to describe directly from video and sentence pairs without having to explicitly identify domain specific objects, actions, and scenes and build classifiers for each. Our models take inspiration from recent breakthroughs in machine translation (Sutskever *et al.*, 2014) and image-captioning (Donahue *et al.*, 2015), and treats the input video as another "language" and translates the visual input to a sequence of words. First, I describe how we can use LSTMs to decode a vector to a sequence of outputs. Next, I present an LSTM based video description model, and also show how we can transfer knowledge from the data rich auxiliary task of image captioning to generate descriptions for short video clips.

3.1 Modeling sequences with LSTMs

Our framework is based on deep image description models in Donahue *et al.* (2015) and Vinyals *et al.* (2015) 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

[1] Based on work published in Venugopalan *et al.* (2015b). All work in this chapter constitutes original contributions.

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:

$$\log p(S|V) = \sum_{t=0}^{N} \log p(S_{w_t}|V, S_{w_1}, \ldots, S_{w_{t-1}}) \tag{3.2}$$

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:

$$h_t = f_\theta(x_t, h_{t-1}) \tag{3.3}$$

where $(h_0 = 0)$. In this work we use the highly successful Long Short-Term Memory (LSTM) network (from Section 2.4.2) as the sequence model, since it has shown superior performance on tasks such as speech recognition (Graves and Jaitly, 2014), machine translation (Sutskever *et al.*, 2014, Cho *et al.*, 2014) and also the more related task of generating sentence descriptions of images (Donahue *et al.*, 2015, Vinyals *et al.*, 2015). Note here that we use LSTM RNNs to specifically decode a vector representation of the video (x_t in Eqn. 3.3) to a sentence. However, we first need to convert videos to a fixed length representation (input x_t), and for this we use a Convolutional Neural Network (CNN). The following sections presents this in greater detail.

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. Our models have been inspired by the use of LSTM RNNs in machine translation (MT) (Sutskever *et al.*, 2014, Bahdanau *et al.*, 2014) where they use one LSTM to encode a sequence of French tokens and another LSTM to decode that encoded representation to a sequence of English words. In contrast to sequence models in traditional statistical MT (Koehn, 2010), RNNs naturally combine with all vector-based representations, such as even those based on images and video. Donahue *et al.* (2015) and Vinyals *et al.* (2015) simultaneously proposed a multimodal analog of the model in Sutskever *et al.* (2014), with an architecture which uses a visual CNN/convnet to encode an image into 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 (Donahue *et al.*, 2014, Zeiler and Fergus, 2014), but require a lot of supervised training data. However, annotated video data with descriptions is scarce. We address this 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 (Krizhevsky *et al.*, 2012). The meaning state and sequence of words is modeled by a recurrent (temporally invariant) deep network pre-trained on 100K+ Flickr (Hodosh *et al.*, 2014) and COCO (Lin *et al.*, 2014) 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 network effectively models the sequence generation task without requiring the use of fixed sentence templates as in previous work (Krishnamoorthy *et al.*, 2013, Guadarrama *et al.*, 2013a, Thomason *et al.*, 2014). Pretraining on image and text data naturally exploits related data to supplement the limited amount of descriptive video currently available. Finally, the use of a deep CNN (Krizhevsky *et al.*, 2012), the winner of the ILSVRC2012 (Russakovsky *et al.*, 2015) image classification competition, provides a strong visual representation of objects, actions and scenes depicted in the video. The main contributions of this

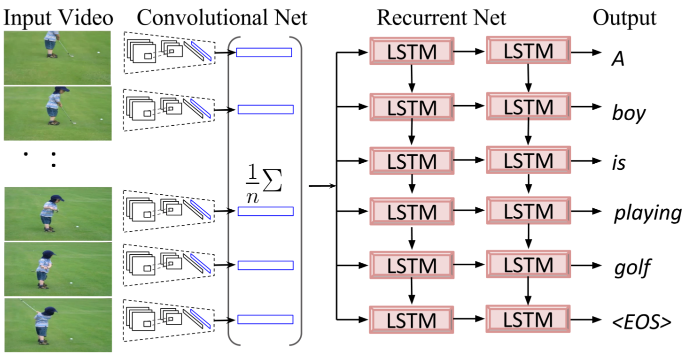


Figure 3.1: The structure of our video description network. We extract visual features from a CNN for each frame, mean pool the features across the entire video and input this at every time step to the LSTM network. The LSTM outputs one word at each time step, based on the video features (and the previous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

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.
- We provide a detailed evaluation of our model on a collection of YouTube videos Chen and Dolan (2011) and demonstrate that it significantly improves over previous state of the art approaches discussed in Chapter 2.

3.3 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 are based on experiments in (Donahue *et al.*, 2017) 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 reference model (Jia *et al.*, 2014), a minor variant of *AlexNet* (Krizhevsky *et al.*, 2012). The net is pre-trained on the 1.2M image ILSVRC-2012 object classification subset of the ImageNet dataset (Russakovsky *et al.*, 2015) 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).

3.3.1 Training and Inference

The two-layer LSTM model is trained to predict the next word S_{w_t} in the sentence given the visual features and the previous $t-1$ words, $p(S_{w_t}|V, S_{w_1}, \ldots, S_{w_{t-1}})$. During training the visual feature, sentence pair (V, S) is provided to the model, which then optimizes the log-likelihood (Equation (7.5)) 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.

$$p(w|z_t) = \frac{\exp(W_w z_t)}{\sum_{w' \in D} \exp(W_{w'} z_t)} \tag{3.4}$$

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.

3.3.2 Architecture and Optimization

In this work, the CNN architecture was based on AlexNet Krizhevsky *et al.* (2012). We used the activations from the $fc - 7$ layer (the penultimate fully-connected layer) of the network. This results in a 4096 dimension feature vector which forms the visual encoding for each frame. Additionally, for each video, we sampled frames at the rate of 3 frames per second. We extract features for the sampled frames which are then averaged (mean-pooled) to form the video encoding. Regarding the LSTM decoder, we use two LSTM layers. Each LSTM layer is unrolled to 20 time-steps, and the hidden dimension of the LSTM is 1000 units. The entire network is trained using stochastic gradient descent with a momentum of 0.9. In addition, we use gradient clipping (set to 5).

3.4 Transfer Learning from Captioned Images

Since the training data available for video description is quite limited (described in Section 3.5), 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 2.4.2, and Section 3.3). As with videos, we extract fc_7 layer features (4096 dimensional vector) from the network (Section 3.3) 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.

3.5 Evaluation

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

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

Models for Comparison

We compare our models against the previous state-of-the-art factor graph model (**FGM**) proposed in Thomason *et al.* (2014) (Section 2.3).

Our LSTM models We present four main models. LSTM-YT is our base two-layer LSTM model trained on just the YouTube video dataset. LSTM-YT$_{flickr}$ is the model trained on the Flickr30k (Hodosh *et al.*, 2014) dataset, and fine tuned on the YouTube dataset as desrcibed in Section 3.4. LSTM-YT$_{coco}$ is first trained on the COCO2014 Lin *et al.* (2014) dataset and then fine-tuned on the video dataset. Our final model, LSTM-YT$_{cocoflickr}$ is trained on the combined data of both the Flickr and COCO models and is tuned on YouTube. 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 by Venugopalan *et al.* (2015b).

3.5.1 SVO Accuracy

Earlier works (Krishnamoorthy *et al.*, 2013, Guadarrama *et al.*, 2013a) that reported results on the YouTube dataset compared their method based on how well their model could predict the subject, verb, and object (SVO) depicted in the video. Since these models first predicted the content (SVO triples) and then generated the sentences, the S,V,O accuracy captured the quality of the content generated by the models. However, in our case the sequential LSTM directly outputs the sentence, so we extract the S,V,O from the dependency parse of the generated sentence. We present, in Table 3.1 and Table 3.2, the accuracy of S,V,O words comparing the performance of our model against any valid ground truth triple and the most frequent triple found in human description for each video. The latter evaluation was also reported by Xu *et al.* (2015c), so we include it here for comparison.

3.5.2 Sentence Generation

To evaluate the generated sentences we use automated Machine Translation metrics BLEU (Papineni *et al.*, 2002) and METEOR (Denkowski and Lavie, 2014) 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 Word-

Model	S%	V%	O%
HVC (Thomason *et al.*, 2014)	86.87	38.66	22.09
FGM (Thomason *et al.*, 2014)	**88.27**	37.16	24.63
LSTM$_{flickr}$	79.95	15.47	13.94
LSTM$_{coco}$	56.30	06.90	14.86
LSTM-YT	79.40	35.52	20.59
LSTM-YT$_{flickr}$	84.92	38.66	21.64
LSTM-YT$_{coco}$	86.58	42.23	**26.69**
LSTM-YT$_{coco+flickr}$	87.27	**42.79**	24.23

Table 3.1: SVO accuracy: Binary SVO accuracy compared against **any valid S,V,O triples** in the ground truth descriptions. We extract S,V,O values from sentences output by our model using a dependency parser. The model is correct if it identifies S,V, or O mentioned in any one of the multiple human descriptions.

Model	S%	V%	O%
HVC (Thomason *et al.*, 2014)	76.57	22.24	11.94
FGM (Thomason *et al.*, 2014)	76.42	21.34	12.39
JointEmbed[2](Xu *et al.*, 2015c)	**78.25**	24.45	11.95
LSTM$_{flickr}$	70.80	10.02	07.84
LSTM$_{coco}$	47.44	02.85	07.05
LSTM-YT	71.19	19.40	09.70
LSTM-YT$_{flickr}$	75.37	21.94	10.74
LSTM-YT$_{coco}$	76.01	23.38	**14.03**
LSTM-YT$_{coco+flickr}$	75.61	**25.31**	12.42

Table 3.2: SVO accuracy: Binary SVO accuracy compared against **most frequent S,V,O triple** in the ground truth descriptions. We extract S,V,O values from parses of sentences output by our model using a dependency parser. The model is correct only if it outputs the most frequently mentioned S, V, O among the human descriptions.

Net synonyms. Image description literature often use BLEU for evaluation, but a more recent study (Elliott and Keller, 2014) 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 3.3. We also present qualitative examples in Figure 3.2 Samples of videos clips with the model's predictions can be found at https:

Model	BLEU	METEOR
FGM (Thomason *et al.*, 2014)	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 3.3: 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 (%).

//www.youtube.com/watch?v=IGaAoW8bA4c.

3.5.3 Training on Individual Frames

Additionally, in order to evaluate the effectiveness of mean pooling, we performed experiments to train and test the model on individual frames from the video. These are presented in Table 3.4. First, we evaluate how well the image description models (i.e. those trained only on images) performed on a randomly sampled frame in the video. These models are denoted as LSTM-YT$_{flickr}$ and LSTM-YT$_{coco}$ in Table 3.4. Next, we used image description models (trained on Flickr30k, COCO or a combination of both) and fine-tuned them on individual frames in the video by picking a different frame for each description in the YouTube dataset. These models, denoted by LSTM-YT-frame$_{flickr}$, LSTM-YT-frame$_{coco}$, LSTM-YT-frame$_{coco+flickr}$ were tested on a random frame from the test video. The overall trends in the results were similar to those seen in Table 3.3. It is also quite evident that the models trained and evaluated on individual frames perform much worse.

3.5.4 Human Evaluation

We used Amazon Mechanical Turk to also collect human judgements. We created a task which employed three Turk workers to watch each video, and rank sentences generated by the different models from “Most Relevant" (5) to “Least Relevant" (1). No two sentences could have the same rank unless they were identi-

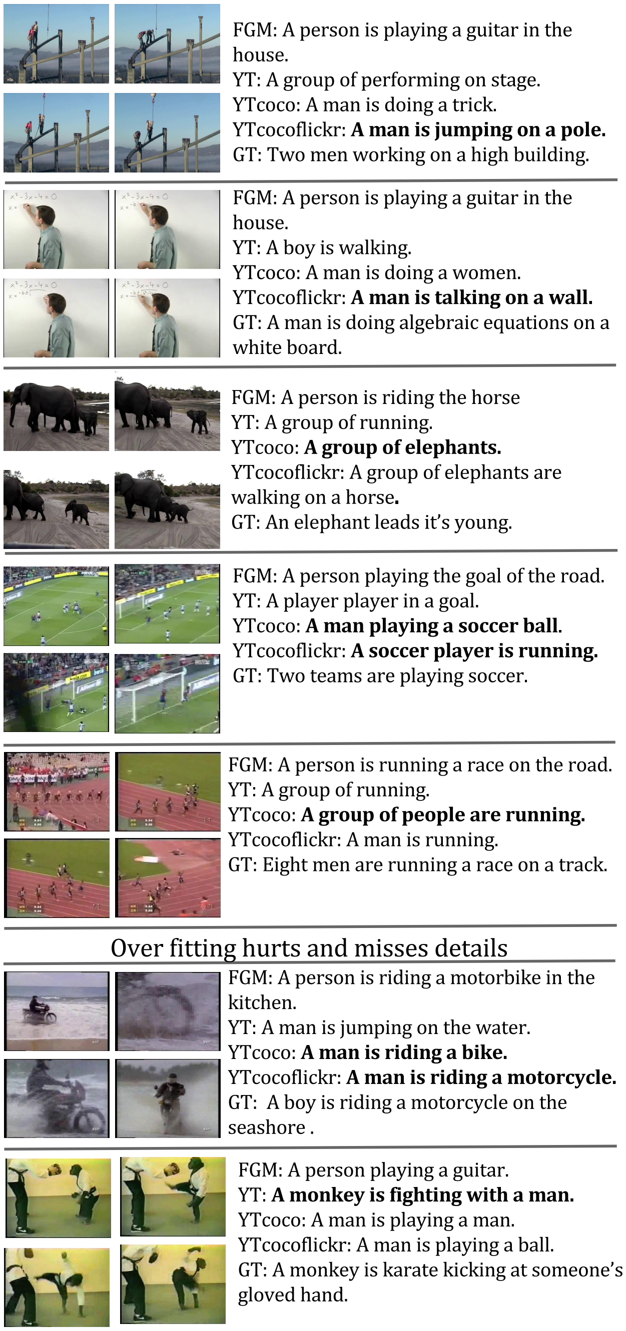
Model (individual frames)	BLEU	METEOR
LSTM$_{flickr}$	08.62	18.56
LSTM$_{coco}$	11.39	20.03
LSTM-YT-frame$_{flickr}$	26.75	26.51
LSTM-YT-frame$_{coco}$	**30.77**	**27.66**
LSTM-YT-frame$_{coco+flickr}$	29.72	27.65

Table 3.4: Scores for BLEU at 4 (combined n-gram 1-4), and METEOR scores comparing the quality of sentence generation by the models trained on Flickr30k and COCO and tested on a random frame from the video. LSTM-YT-frame models were fine tuned on individual frames from the Youtube video dataset. All values are reported as percentage (%).

Model	Relevance	Grammar
FGM (Thomason *et al.*, 2014)	2.26	**3.99**
LSTM-YT	2.74	3.84
LSTM-YT$_{coco}$	**2.93**	3.46
LSTM-YT$_{coco+flickr}$	2.83	3.64
GroundTruth	4.65	4.61

Table 3.5: Human evaluation mean scores. Sentences were uniquely ranked between 1 to 5 based on their relevance to a given video. Sentences were rated between 1 to 5 for grammatical correctness. Higher values are better.

cal. We also evaluate sentences on grammatical correctness. We created a different task which required workers to rate sentences based on grammar. This task displayed only the sentences and did not show any video. Here, workers had to choose a rating between 1-5 for each sentence. Multiple sentences could have the same rating. We discard responses from workers who fail gold-standard items and report the mean ranking/rating for each of the evaluated models in Table 3.5.



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.

3.6 Discussion

From the results in Table 3.3 and the human evaluations in Table 3.5, 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. Also, from Table 3.4 we can see that training on just individual frames in the video is far less effective than mean-pooling frame features across the entire video.

The code and pre-trained models presented in this chapter are made available in the caffe framework and can be downloaded from github[3].

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

Chapter 4

Sequence to Sequence – Video to Text

In this chapter, we propose a novel end-to-end sequence-to-sequence model to generate captions for videos[1]. While the model presented in Chapter 3 only used LSTM based RNNs to decode a visual feature to generate a sentence description, this chapter presents a model that uses an LSTM to both encode a sequence of video frames and decode to generate a sequence of words. A major limitation of our model in Venugopalan *et al.* (2015b) in the previous chapter 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 (Venugopalan *et al.*, 2015a), 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.

4.1 Modeling a sequence of visual inputs

Figure 4.1 depicts our model S2VT. 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 (Brox *et al.*, 2004) 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 (Simonyan and Zisserman, 2014a, Donahue *et al.*, 2015).

To our knowledge, this is the first approach to video description that uses a

[1] Based on work published in Venugopalan *et al.* (2015a). All work in this chapter constitutes original contributions.

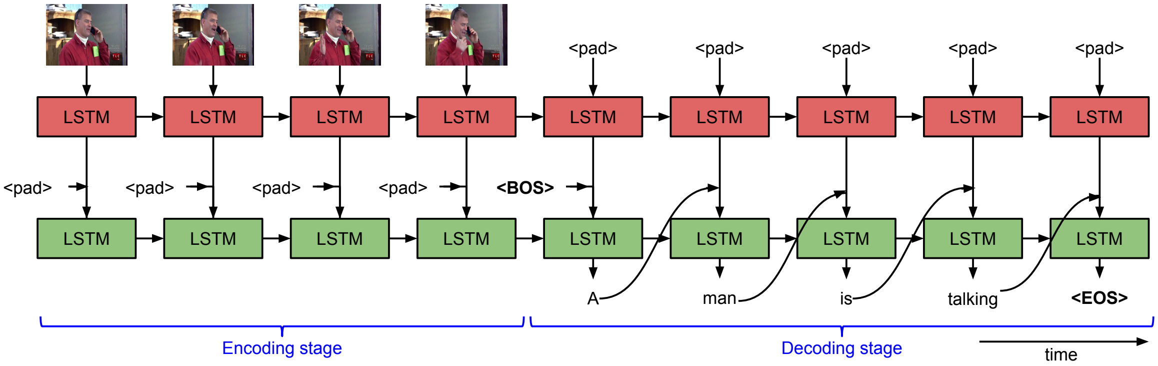


Figure 4.1: S2VT consists of a stack of two LSTMs that learn a representation of a sequence of frames in order to decode it into a sentence that describes the event in the video. The top LSTM layer (colored red) models visual feature inputs (from RGB or optical flow frames). The second LSTM layer (colored 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.

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) (Chen and Dolan, 2011) and two large movie description corpora namely, Montreal Video Annotation Dataset (M-VAD) (Torabi *et al.*, 2015) and MPII Movie Description Dataset (MPII-MD) (Rohrbach *et al.*, 2015b) datasets. We also make our implementation (based on the *Caffe* (Jia *et al.*, 2014) deep learning framework) available on github[2].

4.2 LSTMs for modeling visual and text sequences

Our model uses a stack of two LSTMs with 1000 hidden units each. Figure 4.1 shows the LSTM stack unrolled over time. When two LSTMs are stacked

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

together, as in our case, 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 4.1) 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:

$$\theta^* = \underset{\theta}{\mathrm{argmax}} \sum_{t=1}^{m} \log p(y_t | h_{n+t-1}, y_{t-1}; \theta) \tag{4.1}$$

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:

$$p(y|z_t) = \frac{\exp(W_y z_t)}{\sum_{y' \in V} \exp(W_{y'} z_t)} \tag{4.2}$$

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 (4.2)) until we obtain the <EOS> token.

4.3 Video and text representation

RGB frames. Similar to previous LSTM-based image captioning efforts (Donahue *et al.*, 2015, Vinyals *et al.*, 2015) and our video description approach (Venugopalan *et al.*, 2015b), we forward propagate the input images through a convolutional neural network (CNN) and provide the activations of the penultimate 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 (Simonyan and Zisserman, 2014b). We use CNNs that are pre-trained on the 1.2M image ILSVRC-2012 object classification subset of the ImageNet dataset (Russakovsky *et al.*, 2015) 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 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 (Ng *et al.*, 2015, Donahue *et al.*, 2015) 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 Donahue *et al.* (2015) and Gkioxari and Malik (2015) and first extract classical variational optical flow features (Brox *et al.*, 2004). We then create flow images Gkioxari and Malik (2015), 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 (Gkioxari and Malik, 2015) initialized with weights trained on the UCF101 video dataset (Soomro *et al.*, 2012) 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.

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

Combining RGB and Flow. In our models that combine RGB and Flow input representations of the video, we first train independent S2VT models trained on RGB inputs and Flow inputs. To obtain a combined model, we use a shallow fusion technique to integrate flow and RGB features. At each time step of the decoding phase, the model proposes a set of candidate words. We then re-score these hypotheses with the weighted sum of the scores by the flow and RGB networks, where we only need to recompute the score of each new word $p(y_t = y')$ as:

$$\alpha \cdot p_{rgb}(y_t = y') + (1 - \alpha) \cdot p_{flow}(y_t = y')$$

the hyper-parameter α is tuned on the validation set.

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 4.1). When considering the output of the LSTM we apply a softmax over the complete vocabulary as in Equation (4.2).

4.4 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.

4.4.1 Datasets

In addition to the Microsoft Video Description corpus (MSVD) (Chen and Dolan, 2011) (Section 3.5), we also evaluate our approach on two large movie corpora, namely, the MPII Movie Description Corpus (MPII-MD) (Rohrbach *et al.*, 2015b), and the Montreal Video Annotation Dataset (M-VAD) (Torabi *et al.*, 2015). Statistics of each corpus is presented in Table 7.1.

	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 4.1: 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.

MPII Movie Description Dataset (MPII-MD) MPII-MD (Rohrbach *et al.*, 2015b) 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 (Torabi *et al.*, 2015) 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.

4.4.2 Evaluation Metrics

Quantitative evaluation of the models are performed using the METEOR (Denkowski and Lavie, 2014) 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. (Vedantam *et al.*, 2015) 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 (Chen *et al.*, 2015) to obtain the scores for all our models reported in this section.

4.4.3 Comparisons

We compare our sequence to sequence models against the factor graph model (FGM) in Thomason *et al.* (2014) (Section 2.3), the mean-pooled models (Mean-Pool) in Venugopalan *et al.* (2015b) from the previous chapter and the Soft-Attention models of Yao *et al.* (2015).

The **Soft-Attention** model in Yao *et al.* (2015) 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 (Szegedy *et al.*, 2015) 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.

4.5 Discussion: MSVD dataset

Table 4.2 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 4.2) achieves 27.9% METEOR and improves over the basic mean pooled model proposed in Venugopalan *et al.* (2015b) (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 frames well. When the model is trained with the input frame sequence randomly ordered (line 10 in Table 4.2) the score is considerably lower and comparable to the mean pooled

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

Model	METEOR	
FGM (Thomason *et al.*, 2014)	23.9	(1)
Mean pool		
- AlexNet (Venugopalan *et al.*, 2015b)	26.9	(2)
- VGG	27.7	(3)
- AlexNet COCO pre-trained (Venugopalan *et al.*, 2015b)	29.1	(4)
- GNet (Yao *et al.*, 2015)	28.7	(5)
Soft-attention		
- GoogleNet (Yao *et al.*, 2015)	29.0	(6)
- GoogleNet + 3D-CNN (Yao *et al.*, 2015)	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 4.2: MSVD dataset (METEOR in %, higher is better).

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 11) 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 Yao *et al.* (2015), 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 (Yao *et al.*, 2015) and our S2VT is arguably ours with VGG (line 10) vs. their GoogleNet only model (line 6).

Figure 4.2 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 4.3, 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 cor-

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 4.3: 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 (%).

pus.

Examples showing errors. Figure 4.2 also presents examples where the model makes erroneous predictions. In most cases where there is an error, the model fails to recognize the object or the primary action in the video. Especially if it has never seen such objects or actions during training or has seen them very infrequently. The model also tends to makes grammatical errors such as repeating phrases (on occasion) or spelling mistakes which are actually from misspelled words in the training data.

4.6 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. One of the primary reasons the movie data is far more challenging than the Youtube data is because there is just a single reference translation for each clip. Thus, there is a wider variety of events represented in these clips and far less training data proportional to the variety. This makes both training and evaluation challenging. To avoid over-fitting on the movie corpora we employ drop-out which has proved to be beneficial on these datasets (Rohrbach *et al.*, 2015a). We found it was best to use dropout at the inputs and outputs of both LSTM layers. Further, we used ADAM (Kingma and Ba, 2015) for optimization with a first momentum coefficient of 0.9 and a second momentum coefficient of 0.999. The overall scores on the movie corpora are also much lower due to just a single reference caption during evaluation.

For MPII-MD, reported in Table 4.4, we improve over the SMT approach

Approach (MPII-MD)	METEOR
SMT (best variant) (Rohrbach *et al.*, 2015b)	5.6
Visual-Labels (Rohrbach *et al.*, 2015a)	7.0
Mean pool (VGG)	6.7
S2VT: RGB (VGG), ours	7.1

Table 4.4: MPII-MD dataset (METEOR in %, higher is better).

Approach	METEOR
Visual-Labels (Rohrbach *et al.*, 2015a)	6.3
Temporal attention (Yao *et al.*, 2015)	5.7
Mean pool (VGG)	6.1
S2VT: RGB (VGG), ours	6.7

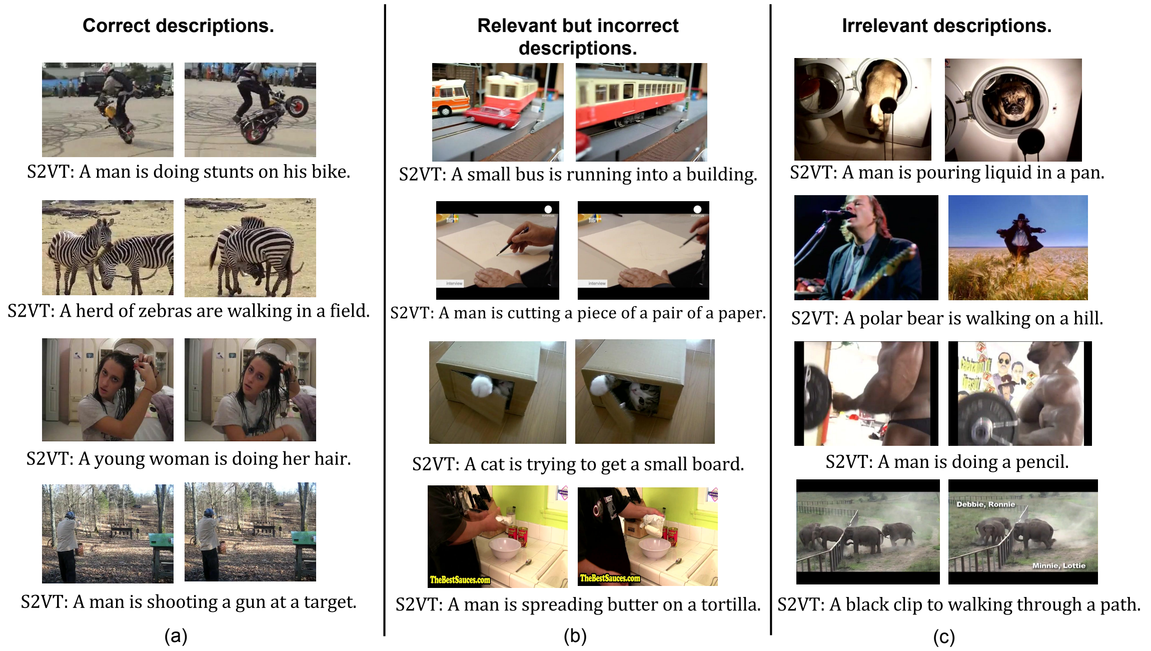
Table 4.5: M-VAD dataset (METEOR in %, higher is better).

from Rohrbach *et al.* (2015b) from 5.6% to 7.1% METEOR and over Mean pooling (Venugopalan *et al.*, 2015b) by 0.4%. Our performance is similar to Visual-Labels (Rohrbach *et al.*, 2015a), 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 (Yao *et al.*, 2015) (5.7%) and Mean pooling (6.1%). On this dataset we also outperform Visual-Labels (Rohrbach *et al.*, 2015a) (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.

In Figure 4.3 we present descriptions generated by our model on some sample clips from the M-VAD dataset.

Discussion on errors. Similar to the errors seen on the Youtube corpus in Figure 4.2, the S2VT model also makes errors on the movie corpora. Looking at some of the descriptions in Figure 4.3, we can notice that while the S2VT model does tend to describe objects/actions in the clip, it doesn't always describe the most salient or important event in the sequence. Comparing descriptions across clips, since the S2VT model describes each clip independently, it does not have any context on the events represented by the previous clips. Hence, it tends to focus on the same event ("driving" in this example) in multiple clips. In Chapter 7 we hope to address these issues by building a model that can describe longer videos taking context into account.


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.

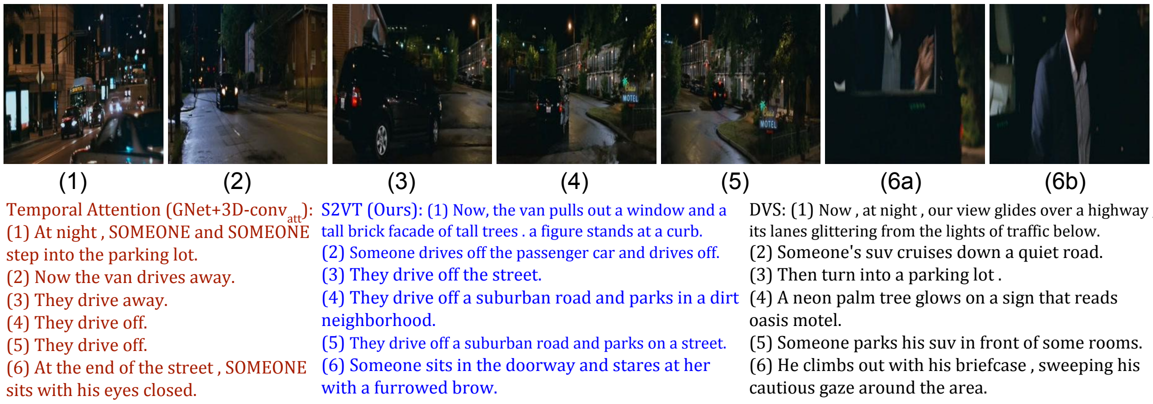
S2VT: A black clip to walking through a path.

(c)

Figure 4.2: 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.

4.7 Summary

Chapters 3 and 4 presented two deep models for video description that used convolutional and recurrent networks, in particular LSTMs to translate from video pixels to sentences. Chapter 3 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 previous works on Youtube clips from the MSVD dataset, and the DVS movie



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 4.3: 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) (Yao *et al.*, 2015), S2VT (in blue) trained on the M-VAD dataset, and DVS: ground truth.

description datasets.

More example video clips, generated sentences, as well as code and models are publicly available[5].

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

Chapter 5

External Knowledge to improve LSTM-based Video Description

While the previous couple of chapters introduced deep recurrent neural network based video description models, this chapter and the next present techniques to improve deep video and image captioning models with knowledge from external sources. This chapter investigates how linguistic knowledge mined from large text corpora can aid the generation of natural language descriptions of videos[1]. Specifically, we integrate both a neural language model and distributional semantics trained on large text corpora into the S2VT video description architecture described previously in Chapter 4. We evaluate our approach on a collection of Youtube videos (the MSVD dataset) as well as the two large movie description datasets MPII-MD and M-VAD showing significant improvements in grammaticality while modestly improving descriptive quality.

5.1 Improving LSTM-based video description with linguistic knowledge mined from text

Deep learning methods such as RNNs need large training corpora; however, there is a lack of high-quality paired video-sentence data. In contrast, raw text corpora are widely available and exhibit rich linguistic structure that can aid video description. Most work in statistical MT utilizes both a language model trained on a large corpus of monolingual target language data as well as a translation model trained on more limited parallel bilingual data. This chapter presents work from Venugopalan *et al.* (2016) which explores methods to incorporate knowledge from language corpora to capture general linguistic regularities to aid video description.

Specifically, this chapter integrates linguistic information into the video-captioning model based on LSTM RNNs from Chapter 4 which have shown state-of-the-art performance on the task. Notably, one of the reason that LSTMs have been successful at captioning is because they are quite effective as language models

[1] Based on work published in Venugopalan *et al.* (2016). All work in this chapter constitutes original contributions.

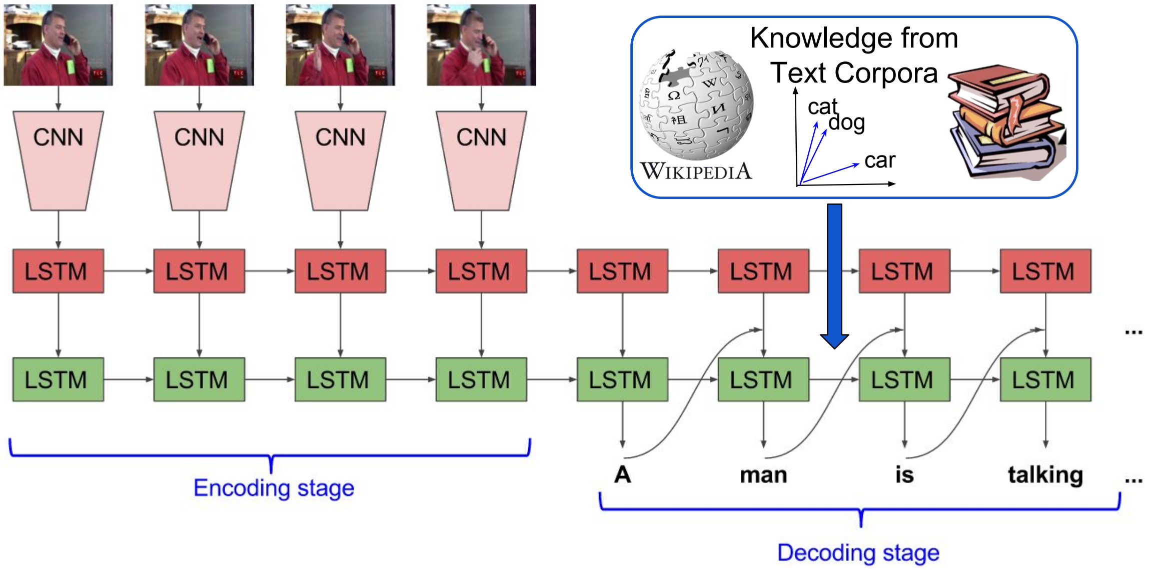


Figure 5.1: The S2VT architecture encodes a sequence of frames and decodes them to a sentence. We propose to add knowledge from text corpora to enhance the quality of video description.

(LMs) (Sundermeyer *et al.*, 2012). Here, we present three approaches to integrate LSTM language models with video captioning models. Our first approach (early fusion) is to pre-train the video description network on plain text before training on parallel video-text corpora. Our next two approaches, inspired by recent MT work (Gulcehre *et al.*, 2015), integrate an LSTM LM with an already trained video-to-text model. Furthermore, we also explore replacing the standard one-hot word encoding of the input words (Section 3.3) with distributional vectors trained on external corpora.

We present detailed comparisons between the approaches, evaluating them on the standard Youtube corpus (Chen and Dolan, 2011) and the two large movie description datasets. The results demonstrate significant improvements in grammaticality of the descriptions (as determined by crowdsourced human evaluations) and more modest improvements in descriptive quality (as determined by both crowdsourced human judgements and standard automated comparison to human-generated descriptions). Our main contributions are

- multiple ways to incorporate knowledge from external text into an existing captioning model,

- extensive experiments comparing the methods on three large video-caption datasets, and
- human judgements to show that external linguistic knowledge has a significant impact on grammar.

5.2 Approaches

Existing visual captioning models (Vinyals *et al.*, 2015, Donahue *et al.*, 2015) are trained solely on text from the caption datasets and tend to exhibit some linguistic irregularities associated with a restricted language model and a small vocabulary. Here, we investigate several techniques to integrate prior linguistic knowledge into a CNN/LSTM-based network for video to text (S2VT) and evaluate their effectiveness at improving the overall description.

5.2.1 Early Fusion

Our first approach (*early fusion*), is to pre-train portions of the network modeling language on large corpora of raw NL text and then continue "fine-tuning" the parameters on the paired video-text corpus. An LSTM model learns to estimate the probability of an output sequence given an input sequence. To learn a language model, we train the LSTM layer to predict the next word given the previous words. Following the S2VT architecture, we embed one-hot encoded words in lower dimensional vectors. The network is trained on web-scale text corpora and the parameters are learned through backpropagation using stochastic gradient descent.[2] The weights from this network are then used to *initialize* the embedding and weights of the LSTM layers of S2VT, which is then trained on video-text data. This trained LM is also used as the LSTM LM in the late and deep fusion models.

[2] The LM was trained to achieve a perplexity of 120

5.2.2 Late Fusion

Our late fusion approach is similar to how neural machine translation models incorporate a trained language model during decoding. At each step of sentence generation, the video caption model proposes a distribution over the vocabulary. We then 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 S2VT 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:

$$\alpha \cdot p_{VM}(y_t = y') + (1 - \alpha) \cdot p_{LM}(y_t = y') \tag{5.1}$$

Hyper-parameter α is tuned on the validation set.

5.2.3 Deep Fusion

In the deep fusion approach (Fig. 5.2), 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 S2VT video description model (h_t^{VM}) and use the combined latent vector to predict the output word. This is similar to the technique proposed by Gulcehre *et al.* (2015) for incorporating language models trained on monolingual corpora for machine translation. However, our approach differs in two key ways: (1) we only concatenate the hidden states of the S2VT LSTM and language LSTM and do not use any additional context information, (2) we fix the weights of the LSTM language model but train the full video captioning network. In this case, the probability of the predicted word at time step t is:

$$p(y_t|\vec{y}_{<t}, \vec{x}) \propto \exp(\mathrm{Wf}(h_t^{VM}, h_t^{LM}) + b) \tag{5.2}$$

where $\vec{x}$ is the visual feature input, W is the weight matrix, and b the biases. We note that, following the same approach by Gulcehre *et al.* (2015) i.e. tuning the LSTM LM, leads to a model that's very similar to the early fusion approach. Hence unlike Gulcehre *et al.* (2015), we avoid tuning the LSTM LM to prevent overwriting already learned weights of a strong language model. However, we do train the full

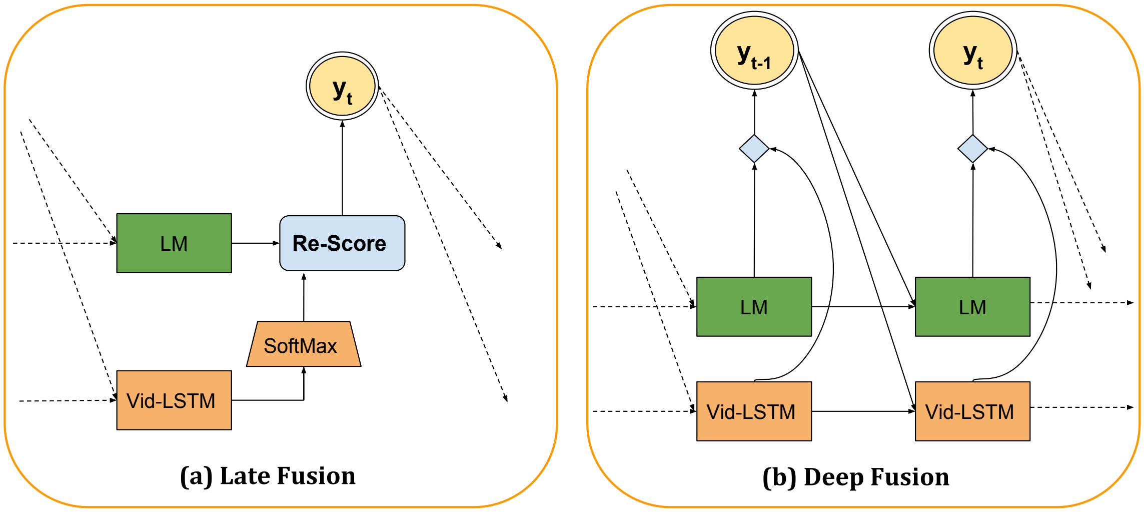


Figure 5.2: Illustration of our late and deep fusion approaches to integrate an independently trained LM to aid video captioning. The deep fusion model learns jointly from the hidden representations of the LM and S2VT video-to-text model (Vid-LSTM), whereas the late fusion re-scores the softmax output of the video-to-text model.

video caption model so that it learns to incorporate the LM outputs while training on the caption data from the paired video-caption domain.

5.2.4 Distributional Word Representations

The S2VT network, like most image and video captioning models, represents words using a 1-of-N (one hot) encoding. During training, the model learns to embed "one-hot" words into a lower 500d space by applying a linear transformation. However, the embedding is learned only from the limited and possibly noisy text in the caption data. There are many approaches (Mikolov *et al.*, 2013, Pennington *et al.*, 2014) that use large text corpora to learn vector-space representations of words that capture fine-grained semantic and syntactic regularities. We propose to take advantage of these to aid video description. Specifically, we replace the embedding matrix from one-hot vectors and instead use 300-dimensional GloVe vectors (Pennington *et al.*, 2014) pre-trained on 6B tokens from Gigaword and Wikipedia 2014. In addition to using the distributional vectors for the input, we also explore variations where the model predicts both the one-hot word (trained on the $\mathrm{softmax}$ loss), as well as predicting the distributional vector from the LSTM hidden state using Euclidean loss as the objective. Here the output vector (y_t) is computed as $y_t = (W_g h_t + b_g)$, and the loss is given by:

$$\mathbb{L}(y_t, w_{glove}) = \|(W_g h_t + b_g) - w_{glove}\|^2 \tag{5.3}$$

where h_t is the LSTM output, w_{glove} is the word's GloVe embedding and W, b are weights and biases. The network then essentially becomes a multi-task model with two loss functions. However, we use this loss only to influence the weights learned by the network, the predicted word embedding is not used.

5.2.5 Ensembling

The overall loss function of the video-caption network is non-convex, and difficult to optimize. In practice, using an ensemble of networks trained slightly differently can improve performance (Hansen and Salamon, 1990). In our work we also present results of an ensemble by averaging the predictions of the best performing models.

5.2.6 Architecture and Optimization

As in Chapter 4, we used the VGG-16 CNN architecture Simonyan and Zisserman (2014b) to encode visual features. With regard to the language model, we used the 300 dimension GloVe embeddings Pennington *et al.* (2014) which form the input to the LSTM network. The language model consisted of a single LSTM layer with a hidden dimension of 1024 units. The video description LSTM consisted of two LSTM layers with a hidden dimension of 1024. The network was trained using the ADAM optimizer Kingma and Ba (2015) with $\alpha = 0.9$ and $\beta = 0.999$. Additionally, we clip gradients over a value of 10.

5.3 Experiments

Datasets. Our language model was trained on sentences from Gigaword, BNC, UkWaC, and Wikipedia. The vocabulary consisted of 72,700 most frequent tokens also containing GloVe embeddings. Following the evaluation in Venugopalan *et al.* (2015a), we compare our models on the collection of Youtube videos in the MSVD dataset (Chen and Dolan, 2011), as well as the two large movie description corpora: MPII-MD (Rohrbach *et al.*, 2015b) and M-VAD (Torabi *et al.*, 2015) described in the previous chapter (Section 4.4.1).

Evaluation Metrics. We evaluate performance using machine translation (MT) metrics METEOR (Denkowski and Lavie, 2014) and BLEU (Papineni *et al.*, 2002) to compare the machine-generated descriptions to human ones. For the movie corpora which have just a single description we use only METEOR which is more robust.

Human Evaluation. We also obtain human judgements using Amazon Turk on a random subset of 200 video clips for each dataset. Each sentence was rated by 3 workers on a Likert scale of 1 to 5 (higher is better) for relevance and grammar. No video was provided during grammar evaluation. For movies, due to copyright, we only evaluate on grammar.

5.3.1 MSVD Dataset Results

On the MSVD dataset, comparison of the proposed techniques in Table 5.1 shows that Deep Fusion performs well on both METEOR and BLEU; incorporating Glove embeddings substantially increases METEOR, and combining them both does best. Our final model is an ensemble (weighted average) of the Glove, and the two Glove+Deep Fusion models trained on the external and in-domain COCO (Lin *et al.*, 2014) sentences. We note here that the state-of-the-art on this dataset is achieved by HRNE (Pan *et al.*, 2016) (METEOR 33.1) which proposes a superior visual processing pipeline using attention to encode the video.

Human ratings also correlate well with the METEOR scores, confirming that our methods give a modest improvement in descriptive quality. However, incorporating linguistic knowledge significantly[3] improves the grammaticality of the results, making them more comprehensible to human users.

[3] Using the Wilcoxon Signed-Rank test, results were significant with $p < 0.02$ on relevance and $p < 0.001$ on grammar.

Model	METEOR	B-4	Relevance	Grammar
S2VT	29.2	37.0	2.06	3.76
Early Fusion	29.6	37.6	-	-
Late Fusion	29.4	37.2	-	-
Deep Fusion	29.6	39.3	-	-
Glove	30.0	37.0	-	-
Glove+Deep				
- Web Corpus	30.3	38.1	2.12	4.05*
- In-Domain	30.3	38.8	2.21*	4.17*
Ensemble	**31.4**	**42.1**	**2.24***	**4.20***
Human	-	-	4.52	4.47

Table 5.1: MSVD dataset: METEOR and BLEU@4 in %, and human ratings (1-5) on relevance and grammar. Best results in bold, * indicates significant over S2VT.

Model	MPII-MD		M-VAD	
	METEOR	Grammar	METEOR	Grammar
S2VT[†]	6.5	2.6	6.6	2.2
Early Fusion	6.7	-	6.8	-
Late Fusion	6.5	-	6.7	-
Deep Fusion	6.8	-	6.8	-
Glove	6.7	3.9*	6.7	3.1*
Glove+Deep	6.8	**4.1***	6.7	**3.3***

Table 5.2: Movie Corpora: METEOR (%) and human grammar ratings (1-5, higher is better). Best results in bold, * indicates significant over S2VT.

Embedding Influence. We experimented multiple ways to incorporate word embeddings: *(1) GloVe input:* Replacing one-hot vectors with GloVe on the LSTM input performed best. *(2) Fine-tuning:* Initializing with GloVe and subsequently fine-tuning the embedding matrix reduced validation results by 0.4 METEOR. *(3) Input and Predict:* Training the LSTM to accept and predict GloVe vectors, as described in Section 5.2.4, performed similar to (1). All scores reported in Tables 5.1 and 5.2 correspond to the setting in (1) with GloVe embeddings used only as input.

S2VT: Someone sits in the bed.
Glove: Someone sits on the couch and watches her phone.
Glove+Deep: Someone sits on the couch, watching her, her feet on her lap.
GT: Someone drops the flowers and kisses someone.

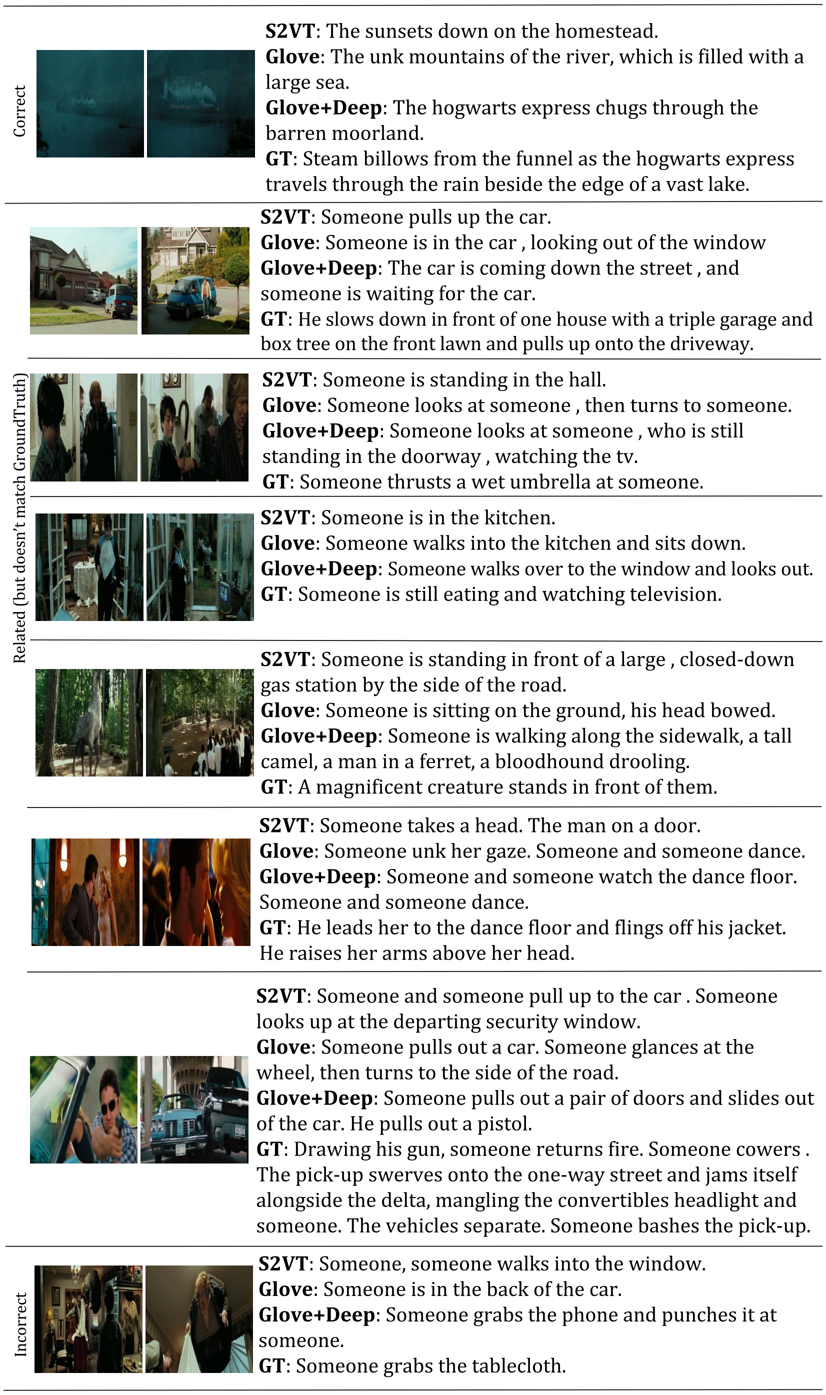
Figure 5.3: Two frames from a clip. Models generate visually relevant sentences but differ from groundtruth (GT).

5.3.2 Movie Description Results

Results on the movie corpora are presented in Table 5.2. Both MPII-MD and M-VAD have only a single ground truth description for each video, which makes both learning and evaluation very challenging (E.g. Fig.5.3). METEOR scores are fairly low on both datasets since generated sentences are compared to a single reference translation. S2VT† is a re-implementation of the base S2VT model with the new vocabulary and architecture (embedding dimension). We observe that the ability of external linguistic knowledge to improve METEOR scores on these challenging datasets is small but consistent. Again, human evaluations show significant (with $p < 0.0001$) improvement in grammatical quality. Figure 5.4 presents more qualitative examples from some movies in the dataset.

5.4 Conclusion

This chapter investigated multiple techniques to incorporate linguistic knowledge from text corpora to aid video captioning. We empirically evaluate our approaches on Youtube clips as well as two movie description corpora. Our results show significant improvements on human evaluations of grammar while modestly improving the overall descriptive quality of sentences on all datasets. While the proposed techniques are evaluated on a specific video-caption network, they are generic and can be applied to other video and image captioning models as we shall see in the next chapter. The code and models from this chapter are shared on http://vsubhashini.github.io/language_fusion.html.



Correct

S2VT: The sunsets down on the homestead.
Glove: The unk mountains of the river, which is filled with a large sea.
Glove+Deep: The hogwarts express chugs through the barren moorland.
GT: Steam billows from the funnel as the hogwarts express travels through the rain beside the edge of a vast lake.

Related (but doesn't match GroundTruth)

S2VT: Someone pulls up the car.
Glove: Someone is in the car , looking out of the window
Glove+Deep: The car is coming down the street , and someone is waiting for the car.
GT: He slows down in front of one house with a triple garage and box tree on the front lawn and pulls up onto the driveway.

S2VT: Someone is standing in the hall.
Glove: Someone looks at someone , then turns to someone.
Glove+Deep: Someone looks at someone , who is still standing in the doorway , watching the tv.
GT: Someone thrusts a wet umbrella at someone.

S2VT: Someone is in the kitchen.
Glove: Someone walks into the kitchen and sits down.
Glove+Deep: Someone walks over to the window and looks out.
GT: Someone is still eating and watching television.

S2VT: Someone is standing in front of a large , closed-down gas station by the side of the road.
Glove: Someone is sitting on the ground, his head bowed.
Glove+Deep: Someone is walking along the sidewalk, a tall camel, a man in a ferret, a bloodhound drooling.
GT: A magnificent creature stands in front of them.

S2VT: Someone takes a head. The man on a door.
Glove: Someone unk her gaze. Someone and someone dance.
Glove+Deep: Someone and someone watch the dance floor. Someone and someone dance.
GT: He leads her to the dance floor and flings off his jacket. He raises her arms above her head.

S2VT: Someone and someone pull up to the car . Someone looks up at the departing security window.
Glove: Someone pulls out a car. Someone glances at the wheel, then turns to the side of the road.
Glove+Deep: Someone pulls out a pair of doors and slides out of the car. He pulls out a pistol.
GT: Drawing his gun, someone returns fire. Someone cowers . The pick-up swerves onto the one-way street and jams itself alongside the delta, mangling the convertibles headlight and someone. The vehicles separate. Someone bashes the pick-up.

Incorrect

S2VT: Someone, someone walks into the window.
Glove: Someone is in the back of the car.
Glove+Deep: Someone grabs the phone and punches it at someone.
GT: Someone grabs the tablecloth.

Figure 5.4: Representative frames from clips in the movie description corpora. S2VT is the baseline model, Glove indicates the model trained with input Glove vectors, and Glove+Deep uses input Glove vectors with the Deep Fusion approach. GT indicates groundtruth sentence.

Chapter 6

External Knowledge for Captioning Novel Objects in Images

Deep captioning models such as the ones presented in the previous chapters are limited in their ability to scale and describe concepts unseen in paired image/video-text corpora. This chapter presents the Novel Object Captioner (NOC), a deep visual semantic captioning model that can describe a large number of object categories not present in existing image-caption datasets[1]. In this chapter, the application of external knowledge is studied with reference to the image-captioning task since labeled images are more widely available (compared to videos) providing us with better scope for training and evaluating our methods. Here, our model takes advantage of external sources – labeled images from object recognition datasets, and semantic knowledge extracted from unannotated text. We propose minimizing a joint objective which can learn from these diverse data sources and leverage distributional semantic embeddings (seen in the previous chapter), enabling the model to generalize and describe novel objects outside of image-caption datasets. We demonstrate that our model exploits semantic information to generate captions for hundreds of object categories in the ImageNet object recognition dataset (Russakovsky *et al.*, 2015) that are not observed in MSCOCO image-caption training data (Lin *et al.*, 2014), as well as many categories that are observed very rarely. Both automatic evaluations and human judgements show that our model considerably outperforms prior work in being able to describe many more categories of objects.

6.1 Captioning images with diverse objects

Modern visual classifiers (He *et al.*, 2016, Simonyan and Zisserman, 2014b) can recognize thousands of object categories, some of which are basic or entry-level (e.g. television), and others that are fine-grained and task specific (e.g. dial-phone, cell-phone). However, recent state-of-the-art visual captioning systems (Donahue *et al.*, 2015, Fang *et al.*, 2015, Karpathy and Fei-Fei, 2015, Kiros *et al.*, 2015, Mao

[1] Based on work published in Venugopalan *et al.* (2017). All work in this chapter constitutes original contributions.

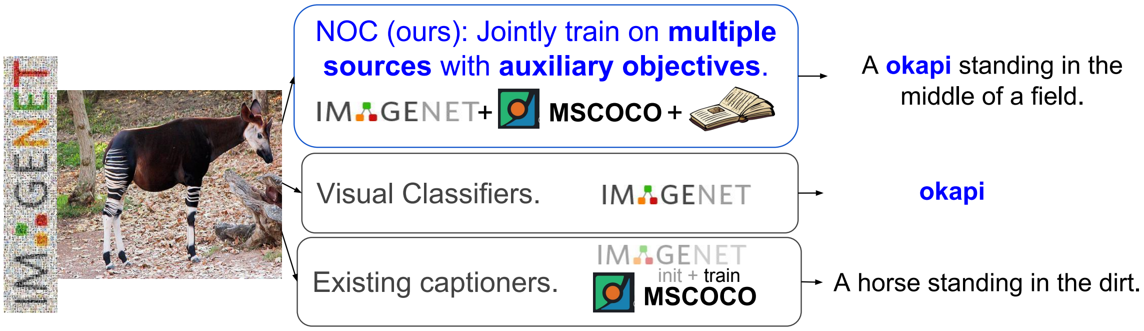

IMAGENET
NOC (ours): Jointly train on multiple sources with auxiliary objectives.
IMAGENET+
MSCOCO +
A okapi standing in the middle of a field.
Visual Classifiers.
IMAGENET
okapi
Existing captioners.
IMAGENET
init + train
MSCOCO
A horse standing in the dirt.

Figure 6.1: We propose a model that learns simultaneously from multiple data sources with auxiliary objectives to describe a variety of objects unseen in paired image-caption data.

et al., 2014, Vinyals *et al.*, 2015) that learn directly from images and descriptions, rely solely on paired image-caption data for supervision and fail in their ability to generalize and describe this vast set of recognizable objects in context. While such systems could be scaled by building larger image/video description datasets, obtaining such captioned data would be expensive and laborious. Furthermore, visual description is challenging because models have to not only correctly identify visual concepts contained in an image, but must also compose these concepts into a coherent sentence.

Recent work (Hendricks *et al.*, 2016) shows that, to incorporate the vast knowledge of current visual recognition networks without explicit paired caption training data, caption models can learn from external sources and learn to compose sentences about visual concepts which are infrequent or non-existent in image-description corpora. However, the pioneering DCC model from Hendricks *et al.* (2016) is unwieldy in the sense that the model requires explicit transfer ("copying") of learned parameters from previously seen categories to novel categories. This not only prevents it from describing rare categories and limits the model's ability to cover a wider variety of objects but also makes it unable to be trained end-to-end. We instead propose the Novel Object Captioner (NOC), a network that can be trained end-to-end using a joint training strategy to integrate knowledge from external visual recognition datasets as well as semantic information from independent unannotated text corpora to generate captions for a diverse range of rare and novel objects (as in Fig. 6.1).

Specifically, we introduce auxiliary objectives which allow our network to learn a captioning model on image-caption pairs simultaneously with a deep language model and visual recognition system on unannotated text and labeled images. Unlike previous work, the auxiliary objectives allow the NOC model to learn relevant information from multiple data sources simultaneously in an end-to-end fashion. Furthermore, NOC implicitly leverages pre-trained distributional word embeddings enabling it to describe unseen and rare object categories. The main contributions of our work are 1) an end-to-end model to describe objects not present in paired image-caption data, 2) auxiliary/joint training of the visual and language models on multiple data sources, and 3) incorporating pre-trained semantic embeddings for the task. We demonstrate the effectiveness of our model by performing extensive experiments on objects held out from MSCOCO (Lin *et al.*, 2014) as well as hundreds of objects from ImageNet (Russakovsky *et al.*, 2015) unseen in caption datasets. Our model substantially outperforms previous work (Hendricks *et al.*, 2016) on both automated as well as human evaluations.

6.2 Novel Object Captioner (NOC) Model

Our NOC model is illustrated in Fig. 6.2. It consists of a language model that leverages distributional semantic embeddings trained on unannotated text and integrates it with a visual recognition model. We introduce auxiliary loss functions (objectives) and jointly train different components on multiple data sources, to create a visual description model which simultaneously learns an independent object recognition model, as well as a language model.

Similar to the language model (LM) seen in the previous chapter (Chapter 5) we start by first training a LSTM-based LM (Sundermeyer *et al.*, 2012) for sentence generation. Again as in Chapter 5 our LM incorporates dense representations for words from distributional embeddings (GloVe, (Pennington *et al.*, 2014)) pre-trained on external text corpora. Simultaneously, we also train a state-of-the-art visual recognition network to provide confidences over words in the vocabulary given an image. This decomposes our model into discrete textual and visual pipelines which can be trained exclusively using unpaired text and unpaired im-

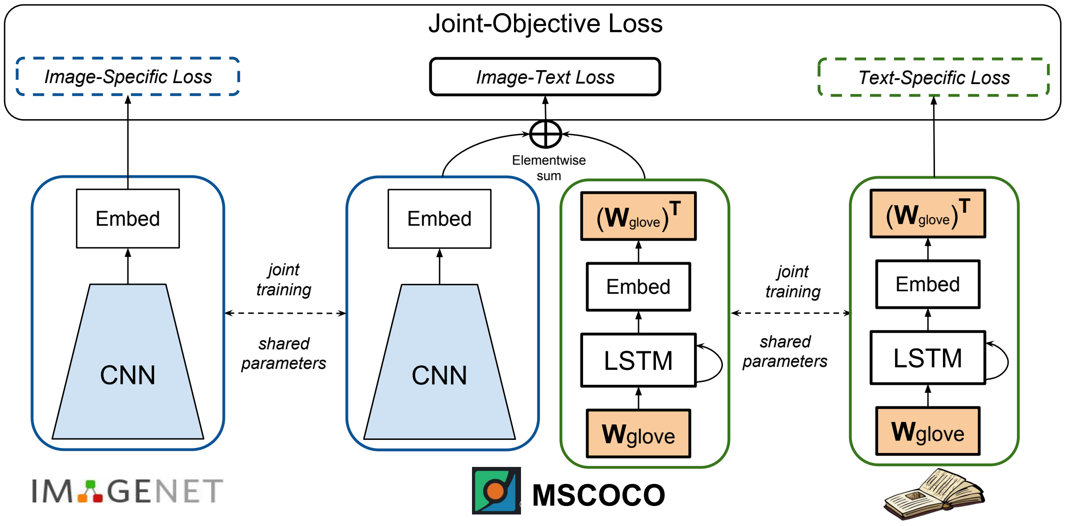


Figure 6.2: Our NOC image caption network. During training, the visual recognition network (left), the LSTM-based language model (right), and the caption model (center) are trained simultaneously on different sources with different objectives but with shared parameters, thus enabling novel object captioning.

age data (networks on left and right of Fig. 6.2). To generate descriptions conditioned on image content, we combine the predictions of our language and visual recognition networks by summing (element-wise) textual and visual confidences over the vocabulary of words. During training, we introduce auxiliary image-specific ($\mathcal{L}_{\mathcal{IM}}$), and text-specific ($\mathcal{L}_{\mathcal{LM}}$) objectives along with the paired image-caption ($\mathcal{L}_{\mathcal{CM}}$) loss. These loss functions, when trained jointly, influence our model to not only produce reasonable image descriptions, but also predict visual concepts as well as generate cohesive text (language modeling). We first discuss the auxiliary objectives and the joint training, and then discuss how we leverage embeddings trained with external text to compose descriptions about novel objects.

6.2.1 Auxiliary Training Objectives

Our motivation for introducing auxiliary objectives is to learn how to describe images without losing the ability to recognize more objects. Typically, image-captioning models incorporate a visual classifier pre-trained on a source domain (e.g. ImageNet dataset) and then tune it to the target domain (the image-caption dataset). However, important information from the source dataset can be suppressed if similar information is not present when fine-tuning, leading the network to forget (over-write weights) for objects not present in the target domain. This is problematic in our scenario in which the model relies on the source datasets to learn a large variety of visual concepts not present in the target dataset. However, with pre-training as well as the complementary auxiliary objectives the model maintains its ability to recognize a wider variety of objects and is encouraged to describe objects which are not present in the target dataset at test time. For the ease of exposition, we abstract away the details of the language and the visual models and first describe the joint training objectives of the complete model, i.e. the text-specific loss, the image-specific loss, and the image-caption loss. We will then describe the language and the visual models.

Image-specific Loss

Our visual recognition model (Fig. 6.2, left) is a neural network parametrized by θ_I and is trained on object recognition datasets. Unlike typical visual recognition models that are trained with a single label on a classification task, for the task of image captioning an image model that has high confidence over multiple visual concepts occurring in an image simultaneously would be preferable. Hence, we choose to train our model using multiple labels (more in Sec. 6.4.2) with a multi-label loss. If l denotes a label and z_l denotes the binary ground-truth value for the label, then the objective for the visual model is given by the cross-entropy loss ($\mathcal{L}_{IM}$):

$$\begin{aligned}\mathcal{L}_{\mathcal{IM}}(I;\theta_I) = -\sum_{l}\Big[& z_l \; log(S_l(f_{IM}(I;\theta_I))) \\ &+ (1-z_l) \; log(1 - S_l(f_{IM}(I;\theta_I)))\Big]\end{aligned} \tag{6.1}$$

where $S_i(x)$ is the output of a $\mathrm{softmax}$ function over index i and input x, and f_{IM}, is the activation of the final layer of the visual recognition network.

Text-specific Loss

Our language model (Fig. 6.2, right) is based on LSTM Recurrent Neural Networks. We denote the parameters of this network by θ_L, and the activation of the final layer of this network by f_{LM}. The language model is trained to predict the next word w_t in a given sequence of words $w_0, ..., w_{t-1}$. This is optimized using the $\mathrm{softmax}$ loss $\mathcal{L}_{\mathcal{LM}}$ which is equivalent to the maximum-likelihood:

$$\begin{aligned} &\mathcal{L}_{\mathcal{LM}}(w_0, ..., w_{t-1}; \theta_L) = \\ &-\sum_t log(S_{w_t}(f_{LM}(w_0, ..., w_{t-1}; \theta_L))) \end{aligned} \tag{6.2}$$

Image-caption Loss

The goal of the image captioning model (Fig. 6.2, center) is to generate a sentence conditioned on an image (I). NOC predicts the next word in a sequence, w_t, conditioned on previously generated words ($w_0, ..., w_{t-1}$) and an image (I), by summing activations from the deep language model, which operates over previous words, and the deep image model, which operates over an image. We denote these final (summed) activations by f_{CM}. Then, the probability of predicting the next word is given by, $P(w_t|w_0, ..., w_{t-1}, I)$

$$\begin{aligned} &=S_{w_t}(f_{CM}(w_0, ..., w_{t-1}, I; \theta)) \\ &=S_{w_t}(f_{LM}(w_0, ..., w_{t-1}; \theta_L) + f_{IM}(I; \theta_I)) \end{aligned} \tag{6.3}$$

Given pairs of images and descriptions, the caption model optimizes the parameters of the underlying language model (θ_L) and image model (θ_I) by minimizing the caption model loss $\mathcal{L}_{\mathcal{CM}} : \mathcal{L}_{\mathcal{CM}}(w_0, ., w_{t-1}, I; \theta_L, \theta_I)$

$$= -\sum_t log(S_{w_t}(f_{CM}(w_0, ., w_{t-1}, I; \theta_L, \theta_I))) \tag{6.4}$$

Joint Training with Auxiliary Losses

While many previous approaches have been successful on image captioning by pre-training the image and language models and tuning the caption model alone (Eqn. 6.4), this is insufficent to generate descriptions for objects outside of the image-caption dataset since the model tends to "forget" (over-write weights) for objects only seen in external data sources. To remedy this, we propose to train the image model, language model, and caption model simultaneously on different data sources. The NOC model's final objective simultaneously minimizes the three individual complementary objectives:

$$\mathcal{L} = \mathcal{L}_{\mathcal{CM}} + \mathcal{L}_{\mathcal{IM}} + \mathcal{L}_{\mathcal{LM}} \tag{6.5}$$

By sharing the weights of the caption model's network with the image network and the language network (as depicted in Fig. 6.2 (a)), the model can be trained simultaneously on independent image-only data, unannotated text data, as well as paired image-caption data. Consequently, co-optimizing different objectives aids the model in recognizing categories outside of the paired image-sentence data.

6.2.2 Language Model with Semantic Embeddings

Our language model here differs in a key aspect compared to the one seen in Chapter 5. Specifically, our language model here consists of the following components: a continuous lower dimensional embedding space for words (W_{glove}), a single recurrent (LSTM) hidden layer (512 dim), and two linear transformation layers where the second layer (W_{glove}^T) maps the vectors to the size of the vocabulary. And the key difference of including a second transformation layer (W_{glove}^T) compared to what we saw in Chapter 5 is particularly crucial for being able to describe novel and unseen object categories as we explain shortly. Finally a softmax activation function is used on the output layer to produce a normalized probability distribution. The cross-entropy loss which is equivalent to the maximum-likelihood is used as the training objective.

In addition to our joint objective (Eqn.6.5), we also employ semantic embeddings in our language model like Venugopalan *et al.* (2016) to help generate sentences when describing novel objects. Specifically, the initial input embedding space (W_{glove}) is used to represent the input (one-hot) words into semantically meaningful dense fixed-length vectors. While the final transformation layer (W_{glove}^T) reverses the mapping (Mao *et al.*, 2014) of a dense vector back to the full vocabulary with the help of a $\mathrm{softmax}$ activation function. These distributional embeddings (Mikolov *et al.*, 2013, Pennington *et al.*, 2014) share the property that words that are semantically similar have similar vector representations. The intuitive reason for using these embeddings in the input and output transformation layers is to help the language model treat words unseen in the image-text corpus to (semantically) similar words that have previously been seen so as to encourage compositional sentence generation i.e. encourage it to use new/rare word in a sentence description based on the visual confidence.

6.2.3 Visual Classifier

The other main component of our model is the visual classifier. Identical to previous work (Hendricks *et al.*, 2016), we employ the VGG-16 (Simonyan and Zisserman, 2014b) convolutional network as the visual recognition network. We modify the final layers of the network to incorporate the multi-label loss (Eqn. 6.1) to predict visual confidence over multiple labels in the full vocabulary. The rest of the classification network remains unchanged.

Finally, we take an elementwise-sum of the visual and language outputs, one can think of this as the language model producing a smooth probability distribution over words (based on GloVe parameter sharing) and then the image signal "selecting" among these based on the visual evidence when summed with the language model beliefs.

6.3 Datasets

In this section we describe the image description dataset as well as the external text and image datasets used in our experiments.

External Text Corpus (WebCorpus) Our external text corpus used to train the language model is identical to the one used in the previous chapter Section 5.3.

Image Caption data To empirically evaluate the ability of NOC to describe new objects we use the training and test set from Hendricks *et al.* (2016). This dataset is created from MSCOCO (Lin *et al.*, 2014) by clustering the main 80 object categories using cosine distance on word2vec (of the object label) and selecting one object from each cluster to hold out from training. The training set holds out images and sentences of 8 objects (bottle, bus, couch, microwave, pizza, racket, suitcase, zebra), which constitute about 10% of the training image and caption pairs in the MSCOCO dataset. Our model is evaluated on how well it can generate descriptions about images containing the eight held-out objects.

Image data We also evaluate sentences generated by NOC on approximately 700 different ImageNet (Russakovsky *et al.*, 2015) objects which are not present in the MSCOCO dataset. We choose this set by identifying objects that are present in both ImageNet and our language corpus (vocabulary), but not present in MSCOCO. Chosen words span a variety of categories including fine-grained categories (e.g., "bloodhound" and "chrysanthemum"), adjectives (e.g., "chiffon", "woollen"), and entry level words (e.g., "toad").

6.4 Evaluation

We empirically evaluate the ability of our proposed model to describe novel objects by following the experimental setup of Hendricks *et al.* (2016). We first evaluate our model on the set of held out objects from the MSCOCO dataset as described in Section 6.3. To demonstrate scalability, we also evaluate on the ImageNet dataset (Deng *et al.*, 2009b). We perform several additional experiments with different ablations of the models, as well as comparisons with training on in-domain and out-of-domain data, and evaluations on objects that are infrequent/rare in image-caption data, these results can be found in the original paper (Venugopalan *et al.*, 2017).

6.4.1 Evaluation on MSCOCO

We perform the following experiments to compare NOC's performance with previous work (Hendricks *et al.*, 2016): 1. We evaluate the model's ability to caption objects that are held out from MSCOCO during training (Sec. 6.4.2). 2. To study the effect of the data source on training, we report performance of NOC when the image and language networks are trained on in-domain and out-of-domain sources (Sec. 6.4.3). In addition to these, to understand our model better: 3. We perform ablations to study how much each component of our model (such as word embeddings, auxiliary objective, etc.) contributes to the performance (Sec. 6.4.4). 4. We also study if the model's performance remains consistent when holding out a different subset of objects from MSCOCO (Sec. 6.4.5).

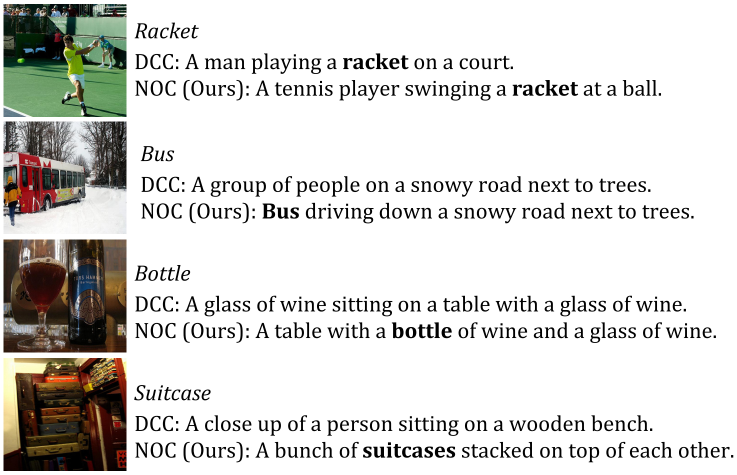
6.4.2 Captioning held-out objects

Following the setup in Hendricks *et al.* (2016), we evaluate our model's ability to caption objects that are held out from MSCOCO during training. We optimize each loss in our model with the following datasets: the caption model, which jointly learns the parameters θ_L and θ_I, is trained only on the subset of MSCOCO without the 8 objects (see section 6.3), the image model, which updates parameters θ_I, is optimized using labeled images, and the language model which updates parameters θ_L, is trained using the corresponding descriptions. When training the visual network on images from COCO, we obtain multiple labels for each image by considering all words in the associated captions as labels after removing stop words. We use the METEOR metric (Denkowski and Lavie, 2014) to evaluate description quality. However, METEOR only captures fluency and does not account for the mention (or lack) of specific words. Hence, we also use F1 to ascertain that the model mentions the object name in the description of the images containing the object. Thus, the metrics measure if the model can both identify the object and use it fluently in a sentence.

Metric	Model	bottle	bus	couch	microwave	pizza	racket	suitcase	zebra	Avg.
F1	DCC	4.63	29.79	**45.87**	**28.09**	64.59	52.x24	13.16	79.88	39.78
	NOC	**17.78**	**68.79**	25.55	24.72	**69.33**	**55.31**	**39.86**	**89.02**	**48.79**
MET.	DCC	18.1	**21.6**	**23.1**	22.1	22.2	20.3	18.3	22.3	21.00
	NOC	**21.2**	20.4	21.4	21.5	21.8	**24.6**	18.0	21.8	**21.32**

Table 6.1: MSCOCO Captioning: F1 and METEOR (denoted MET.) scores (in %) of NOC (our model) and DCC (Hendricks *et al.*, 2016) on held-out objects not seen jointly during image-caption training, along with the average scores of the generated captions across images containing these objects.

COCO heldout objects. Table 6.1 compares the F1 score achieved by NOC to the previous best method, DCC (Hendricks *et al.*, 2016) on the 8 held-out COCO



Racket
DCC: A man playing a **racket** on a court.
NOC (Ours): A tennis player swinging a **racket** at a ball.

Bus
DCC: A group of people on a snowy road next to trees.
NOC (Ours): **Bus** driving down a snowy road next to trees.

Bottle
DCC: A glass of wine sitting on a table with a glass of wine.
NOC (Ours): A table with a **bottle** of wine and a glass of wine.

Suitcase
DCC: A close up of a person sitting on a wooden bench.
NOC (Ours): A bunch of **suitcases** stacked on top of each other.

Figure 6.3: COCO Captioning: Examples comparing captions by NOC (ours) and DCC (Hendricks *et al.*, 2016) on held out objects from MSCOCO.

objects. NOC outperforms DCC (by 10% F1 on average) on all objects except "couch" and "microwave". The higher F1 and METEOR demonstrate that NOC is able to correctly recognize many more instances of the unseen objects and also integrate the words into fluent descriptions. Figure 6.3 presents some qualitative examples comparing the two models.

6.4.3 Training data source

To study the effect of different data sources, we also evaluate our model in an out-of-domain setting where classifiers for held out objects are trained with images from ImageNet and the language model is trained on text mined from external corpora. Table 6.2 reports average scores across the eight held-out objects. We compare our NOC model to results from Hendricks *et al.* (2016) (DCC), as well as a competitive image captioning model - LRCN (Donahue *et al.*, 2015) trained on the same split. In the out-of-domain setting (line 2), for the chosen set of 8 objects, NOC performs slightly better on F1 and a bit lower on METEOR compared to DCC. However, as previously mentioned, DCC needs to explicitly identify a set of "seen" object classes to transfer weights to the novel classes whereas NOC can be used for inference directly. DCC's transfer mechanism also leads to peculiar descriptions. E.g., *Racket* in Fig. 6.3.

	Image	Text	Model	METEOR	F1
1	Baseline (no transfer)		LRCN	19.33	0
			DCC	19.90	0
2	Image Net	Web Corpus	DCC	20.66	34.94
			NOC	17.56	36.50
3	COCO	Web Corpus	NOC	19.18	41.74
4	COCO	COCO	DCC	21.00	39.78
			NOC	**21.32**	**48.79**

Table 6.2: Comparison with different training data sources on 8 held-out COCO objects. Having in-domain data helps both the DCC (Hendricks *et al.*, 2016) and our NOC model caption novel objects.

With COCO image training (line 3), F1 scores of NOC improves considerably even with the Web Corpus LM training. Finally in the in-domain setting (line 4) NOC outperforms DCC on F1 by around 10 points while also improving METEOR slightly. This suggests that NOC is able to associate the objects with captions better with in-domain training, and the auxiliary objectives and embedding help the model to generalize and describe novel objects.

6.4.4 Ablations

Table 6.3 compares how different aspects of training impact the overall performance. *Tuned Vision contribution:* The model that does not incorporate Glove or LM pre-training has poor performance (METEOR 15.78, F1 14.41); this ablation shows the contribution of the vision model alone in recognizing and describing the held out objects. *LM & Glove contribution:* The model trained without the auxiliary objective, performs better with F1 of 25.38 and METEOR of 19.80; this improvement comes largely from the GloVe embeddings which help in captioning novel object classes. *LM & Pre-trained Vision:* It's interesting to note that when we fix classifier's weights (pre-trained on all objects), before tuning the LM on the image-caption COCO subset, the F1 increases substantially to 39.70 suggesting that the visual model recognizes many objects but can "forget" objects learned by the classifier when fine-tuned on the image-caption data (without the 8 objects). *Auxiliary*

Contributing factor	Glove	LM pretrain	Tuned CNN	Auxiliary Objective	METEOR	F1
Tuned Vision	-	-	✓	✓	15.78	14.41
LM & Embedding	✓	✓	✓	-	19.80	25.38
LM & Pre-trained Vision	✓	✓	Fixed	-	18.91	39.70
Auxiliary Objective	✓	-	✓	✓	19.69	47.02
All	✓	✓	✓	✓	**21.32**	**48.79**

Table 6.3: Ablations comparing the contributions of the Glove embedding, LM pre-training, and auxiliary objectives, of the NOC model. Our auxiliary objective along with Glove have the largest impact in captioning novel objects.

Objective: Incorporating the auxiliary objectives, F1 improves remarkably to 47.02. We note here that by virtue of including auxiliary objectives the visual network is tuned on all images thus retaining it's ability to classify/recognize a wide range of objects. Finally, incorporating all aspects gives NOC the best performance (F1 48.79, METEOR 21.32), significantly outperforming DCC.

6.4.5 Validating on a different subset of COCO

To show that our model is consistent across objects, we create a different training/test split by holding out a different set of eight objects from COCO. The objects we hold out are: bed, book, carrot, elephant, spoon, toilet, truck and umbrella. Images and sentences from these eight objects again constitute about 10% of the MSCOCO training dataset. Table 6.4 presents the performance of the model on this subset. We observe that the F1 and METEOR scores, although a bit lower, are consistent with numbers observed in Table 6.1 confirming that our model is able to generalize to different subsets of objects.

6.4.6 Scaling to ImageNet

To demonstrate the scalability of NOC, we describe objects in ImageNet for which no paired image-sentence data exists. Our experiments are performed on two subsets of ImageNet, (i) Novel Objects: A set of 638 objects which are present in ImageNet as well as the model's vocabulary but are not mentioned in MSCOCO.

Model	bed	book	carrot	elephant	spoon	toilet	truck	umbrella	Av. F1	M.
NOC	53.31	18.58	20.69	85.35	02.70	73.61	57.90	54.23	45.80	20.04

Table 6.4: MSCOCO Captioning: F1 scores (in %) of NOC (our model) on a different subset of the held-out objects not seen jointly during image-caption training, along with the average F1 and METEOR scores (denoted M.) of the generated captions across images containing these objects. NOC is consistently able to caption different subsets of unseen object categories in MSCOCO.

(ii) Rare Objects: A set of 52 objects which are in ImageNet as well as the MSCOCO vocabulary but are mentioned infrequently in the MSCOCO captions (median of 27 mentions). For quantitative evaluation, (i) we measure the percentage of objects for which the model is able to describe at least one image of the object (using the object label), (ii) we also report accuracy and F1 scores to compare across the entire set of images and objects the model is able to describe. Furthermore, we obtain human evaluations comparing our model with previous work on whether the model is able to incorporate the object label meaningfully in the description together with how well it describes the image.

6.4.7 Describing Novel Objects

Table 6.5 compares models on 638 novel object categories (identical to Hendricks *et al.* (2016)) using the following metrics: (i) Describing novel objects (%) refers to the percentage of the selected ImageNet objects mentioned in descriptions, i.e. for each novel word (e.g., “otter”) the model should incorporate the word (“otter”) into at least one description about an ImageNet image of the object (otter). While DCC is able to recognize and describe 56.85% (363) of the selected ImageNet objects in descriptions, NOC recognizes several more objects and is capable of describing 91.27% (582 of 638) ImageNet objects. (ii) Accuracy refers to the percentage of images from each category where the model is able to correctly identify and describe the category. We report the average accuracy across all categories. DCC incorporates a new word correctly 11.08% of the time, in comparison, NOC improves this appreciably to 24.74%. (iii) F1 score is computed based on precision and recall of mentioning the object in the description. Again, NOC outperforms

Model	Desc. Novel (%)	Acc (%)	F1 (%)
DCC	56.85	11.08	14.47
NOC	**91.27**	**24.74**	**33.76**

Table 6.5: ImageNet: Comparing our model against DCC (Hendricks *et al.*, 2016) on % of novel classes described, average accuracy of mentioning the class in the description, and mean F1 scores for object mentions.

with average F1 33.76% to DCC's 14.47%.

Although NOC and DCC (Hendricks *et al.*, 2016) use the same CNN, NOC is both able to describe more categories, and correctly integrate new words into descriptions more frequently. DCC (Hendricks *et al.*, 2016) can fail either with respect to finding a suitable object that is both semantically and syntactically similar to the novel object, or with regard to their language model composing a sentence using the object name, in NOC the former never occurs (i.e. we don't need to explicitly identify similar objects), reducing the overall sources of error. Qualitative examples of images and sentences generated by the models are presented in Figure 6.5.

Fig. 6.5 and Fig. 6.4 (column 3) show examples where NOC describes a large variety of objects from ImageNet. Fig. 6.5 compares our model with DCC. Fig. 6.6 and Fig. 6.4 (right) outline some errors. Failing to describe a new object is one common error for NOC. E.g. Fig. 6.4 (top right), NOC incorrectly describes a man holding a "sitar" as a man holding a "baseball bat". Other common errors include generating non-grammatical or nonsensical phrases (example with "gladiator", "aardvark") and repeating a specific object ("A barracuda ... with a barracuda", "trifle cake").

6.4.8 Describing Rare Objects/Words

In order to understand if including additional data enables NOC to better generate sentences about rare words, we select 52 objects which are in ImageNet as well as the MSCOCO vocabulary. The selected rare words occur with varying frequency in the MSCOCO training set, with about 52 mentions on average (median 27) across all training sentences. For example, words such as "bonsai" only appear 5 times,"whisk" (11 annotations), "teapot" (30 annotations), and others such as pumpkin appears 58 times, "swan" (60 annotations), and on the higher side

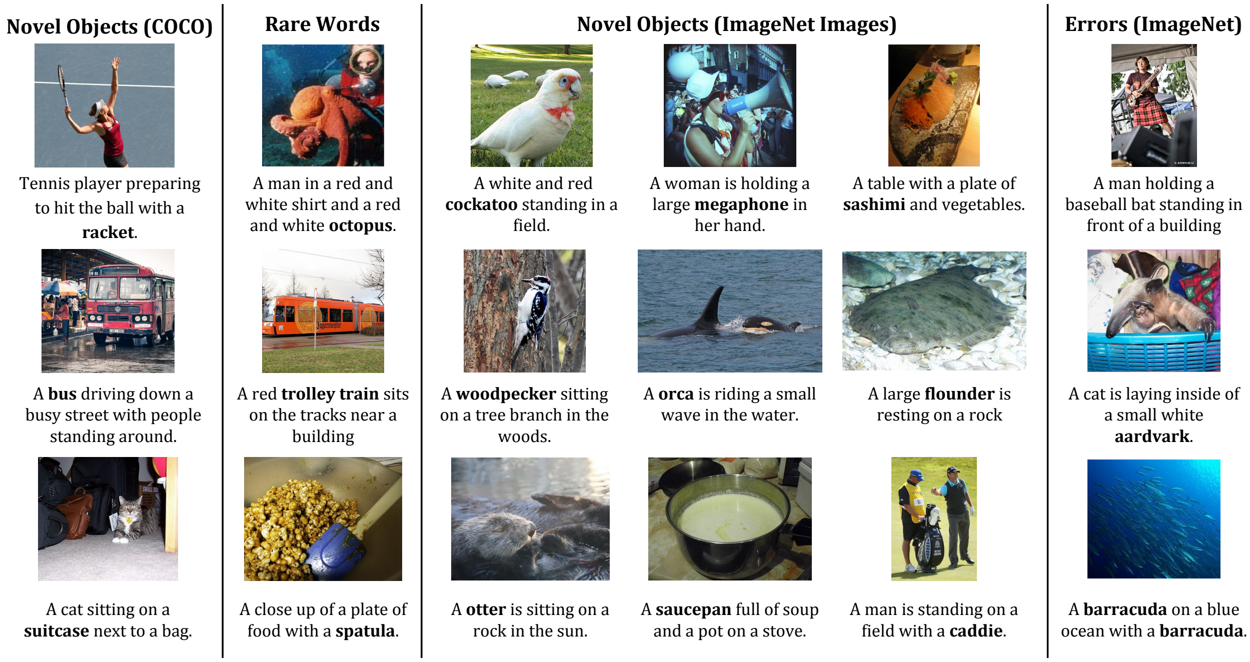
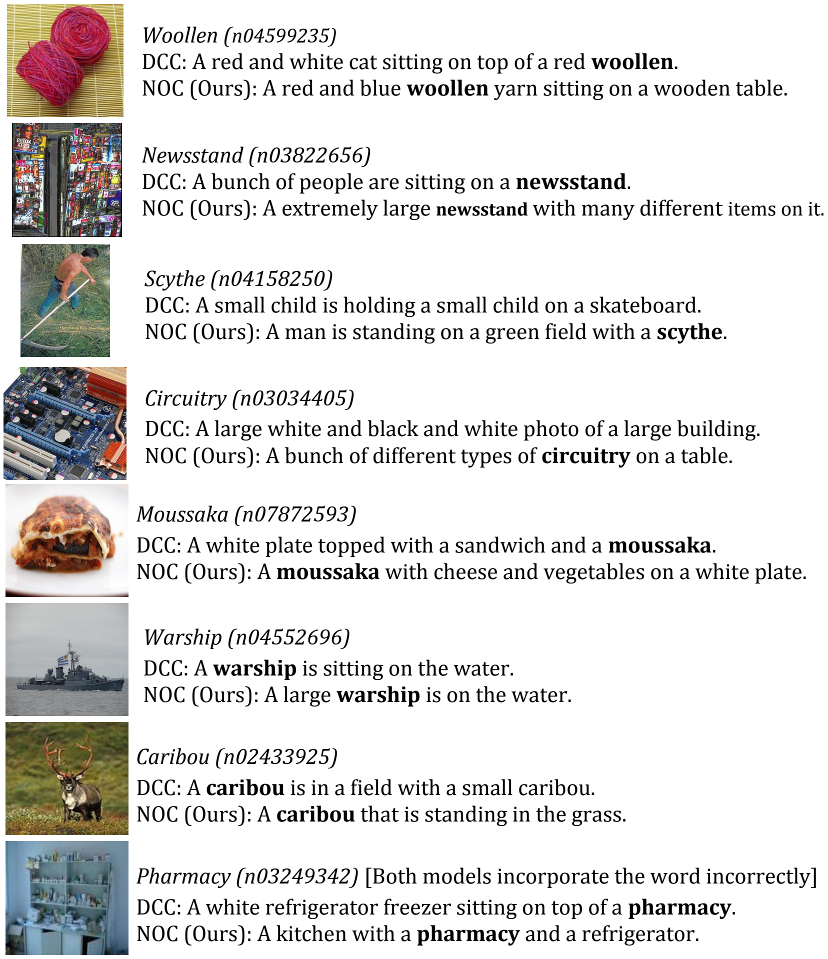

Novel Objects (COCO)
Rare Words
Novel Objects (ImageNet Images)
Errors (ImageNet)
Tennis player preparing to hit the ball with a **racket**.
A man in a red and white shirt and a red and white **octopus**.
A white and red **cockatoo** standing in a field.
A woman is holding a large **megaphone** in her hand.
A table with a plate of **sashimi** and vegetables.
A man holding a baseball bat standing in front of a building
A **bus** driving down a busy street with people standing around.
A red **trolley train** sits on the tracks near a building
A **woodpecker** sitting on a tree branch in the woods.
A **orca** is riding a small wave in the water.
A large **flounder** is resting on a rock
A cat is laying inside of a small white **aardvark**.
A cat sitting on a **suitcase** next to a bag.
A close up of a plate of food with a **spatula**.
A **otter** is sitting on a rock in the sun.
A **saucepan** full of soup and a pot on a stove.
A man is standing on a field with a **caddie**.
A **barracuda** on a blue ocean with a **barracuda**.

Figure 6.4: Descriptions produced by NOC on a variety of objects, including "caddie", "saucepan", and "flounder". (Right) NOC makes errors and (top right) fails to describe the new object ("sitar").

objects like scarf appear 144 times. When tested on ImageNet images containing these concepts, a model trained only with MSCOCO paired data incorporates rare words into sentences 2.93% of the time with an average F1 score of 4.58%. In contrast, integrating outside data, our NOC model can incorporate rare words into descriptions 35.15% of the time with an average F1 score of 47.58%. We do not compare this to DCC since DCC cannot be applied directly to caption rare objects.

6.4.9 Human Evaluation

ImageNet images do not have accompanying captions and this makes the task much more challenging to evaluate. To compare the performance of NOC and DCC we obtain human judgements on captions generated by the models on several object categories. We select 3 images each from about 580 object categories that at least one of the two models, NOC and DCC, can describe. (Note that although both models were trained on the same ImageNet object categories, NOC is able to describe almost all of the object categories that have been described by



Woollen (n04599235)
DCC: A red and white cat sitting on top of a red **woollen**.
NOC (Ours): A red and blue **woollen** yarn sitting on a wooden table.

Newsstand (n03822656)
DCC: A bunch of people are sitting on a **newsstand**.
NOC (Ours): A extremely large **newsstand** with many different items on it.

Scythe (n04158250)
DCC: A small child is holding a small child on a skateboard.
NOC (Ours): A man is standing on a green field with a **scythe**.

Circuitry (n03034405)
DCC: A large white and black and white photo of a large building.
NOC (Ours): A bunch of different types of **circuitry** on a table.

Moussaka (n07872593)
DCC: A white plate topped with a sandwich and a **moussaka**.
NOC (Ours): A **moussaka** with cheese and vegetables on a white plate.

Warship (n04552696)
DCC: A **warship** is sitting on the water.
NOC (Ours): A large **warship** is on the water.

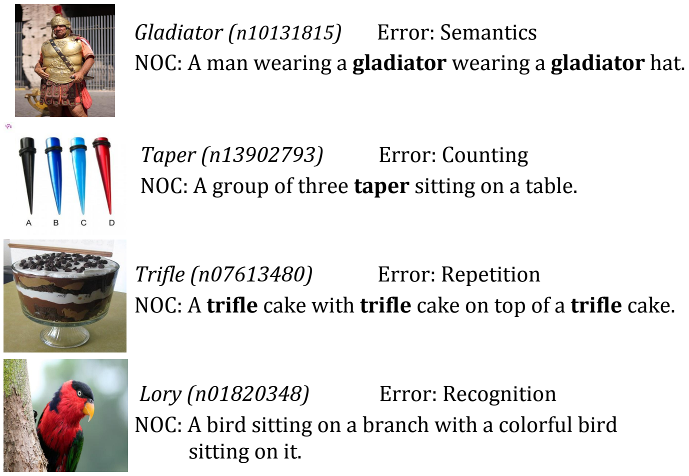
Caribou (n02433925)
DCC: A **caribou** is in a field with a small caribou.
NOC (Ours): A **caribou** that is standing in the grass.

Pharmacy (n03249342) [Both models incorporate the word incorrectly]
DCC: A white refrigerator freezer sitting on top of a **pharmacy**.
NOC (Ours): A kitchen with a **pharmacy** and a refrigerator.

Figure 6.5: ImageNet Captioning: Examples comparing captions by NOC (ours) and DCC (Hendricks *et al.*, 2016) on objects from ImageNet.

DCC). When selecting the images, for object categories that both models can describe, we make sure to select at least two images for which both models mention the object label in the description. Each image is presented to three workers. We conducted two human studies: Given the image, the ground-truth object category (and meaning), and the captions generated by the models, we evaluate on:

Word Incorporation: We ask humans to choose which sentence/caption incorporates the object label meaningfully in the description. The options provided are: (i) Sentence 1 incorporates the word better, (ii) Sentence 2 incorporates the word better, (iii) Both sentences incorporate the word equally well, or (iv)



Gladiator (n10131815) Error: Semantics
NOC: A man wearing a **gladiator** wearing a **gladiator** hat.

Taper (n13902793) Error: Counting
NOC: A group of three **taper** sitting on a table.

Trifle (n07613480) Error: Repetition
NOC: A **trifle** cake with **trifle** cake on top of a **trifle** cake.

Lory (n01820348) Error: Recognition
NOC: A bird sitting on a branch with a colorful bird sitting on it.

Figure 6.6: ImageNet Captioning: Common types of errors observed in the captions generated by the NOC model.

Neither of them do well.

Image Description: We also ask humans to pick which of the two sentences describes the image better.

This allows us to compare both how well a model incorporates the novel object label in the sentence, as well as how appropriate the description is to the image. The results are presented in Table 6.6. On the subset of images corresponding to objects that both models can describe (Intersection), NOC and DCC appear evenly matched, with NOC only having a slight edge. However, looking at all object categories (Union), NOC is able to both incorporate the object label in the sentence, and describe the image better than DCC.

6.5 ImageNet Qualitative Examples

We present additional examples of the NOC model's descriptions on Imagenet images. We first present some examples where the model is able to generate descriptions of an object in different contexts. Then we present several examples to demonstrate the diversity of objects that NOC can describe. We then present examples where the model generates erroneous descriptions and categorize these errors.

	Word Incorporation		Image Description	
Objects subset →	Union	Intersection	Union	Intersection
NOC is better	**43.78**	34.61	**59.84**	51.04
DCC is better	25.74	34.12	40.16	48.96
Both equally good	6.10	9.35	-	
Neither is good	24.37	21.91	-	

Table 6.6: ImageNet: Human judgements comparing our NOC model with DCC (Hendricks *et al.*, 2016) on the ability to meaningfully incorporate the novel object in the description (Word Incorporation) and describe the image. 'Union' and 'Intersection' refer to the subset of objects where atleast one model, and both models are able to incorporate the object name in the description. All values in %.

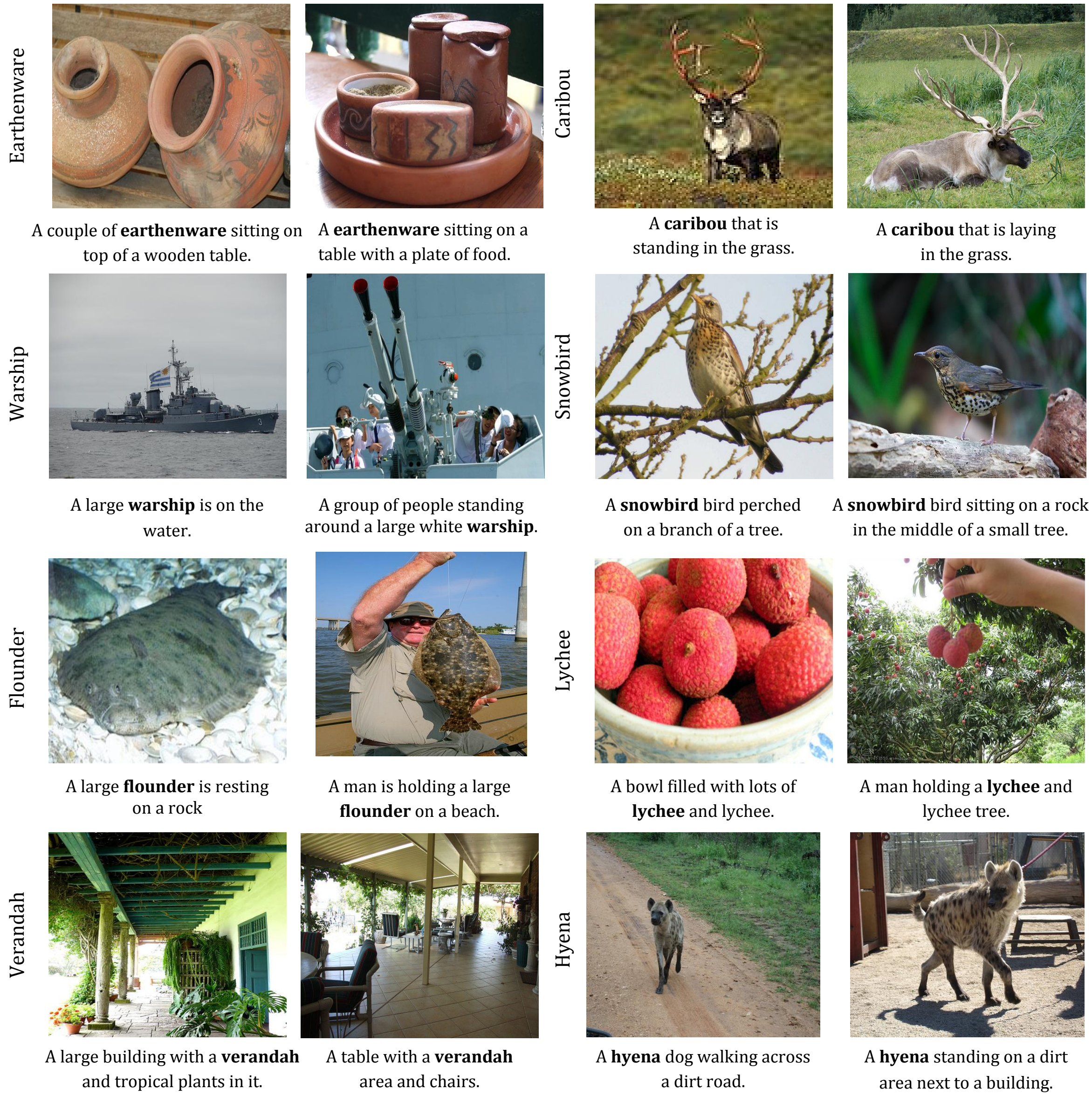
6.5.1 Context

Fig. 6.7 shows images of eight objects, each in two different settings from ImageNet. Images show objects in different backgrounds (Snowbird on a tree branch and on a rock, Hyena on a dirt path and near a building); actions (Caribou sitting vs lying down); and being acted upon differently (Flounder resting and a person holding the fish, and Lychees in a bowl vs being held by a person). NOC is able to capture the context information correctly while describing the novel objects (eartherware, caribou, warship, snowbird, flounder, lychee, verandah, and hyena).

6.5.2 Object Diversity

Fig. 6.8 and Fig. 6.9 present descriptions generated by NOC on a variety of object categories such as birds, animals, vegetable/fruits, food items, household objects, kitchen utensils, items of clothing, musical instruments, indoor and outdoor scenes among others. While almost all novel words (nouns in Imagenet) correspond to objects, NOC learns to use some of them more appropriately as adjectives ('chiffon' dress in Fig. 6.8, 'brownstone' building and 'tweed' jacket in Fig. 6.9 as well as 'woollen' yarn in Fig. 6.5.

Comparison with prior work. Additionally, for comparison with the DCC model from Hendricks *et al.* (2016), Fig. 6.9 presents images of objects that both models can describe, and captions generated by both DCC and NOC.



A couple of **earthenware** sitting on top of a wooden table.

A **earthenware** sitting on a table with a plate of food.

A **caribou** that is standing in the grass.

A **caribou** that is laying in the grass.

A large **warship** is on the water.

A group of people standing around a large white **warship**.

A **snowbird** bird perched on a branch of a tree.

A **snowbird** bird sitting on a rock in the middle of a small tree.

A large **flounder** is resting on a rock

A man is holding a large **flounder** on a beach.

A bowl filled with lots of **lychee** and lychee.

A man holding a **lychee** and lychee tree.

A large building with a **verandah** and tropical plants in it.

A table with a **verandah** area and chairs.

A **hyena** dog walking across a dirt road.

A **hyena** standing on a dirt area next to a building.

Figure 6.7: Examples showing descriptions generated by NOC for ImageNet images of eight objects, each in two different contexts. NOC is often able to generate descriptions incorporating both the novel object name as well as the background context correctly.

A small **pheasant** is standing in a field.

A **osprey** flying over a large grassy area.

A large **glacier** with a mountain in the background.

A group of people are sitting in a **baobab**.

A **humpback** is flying over a large body of water.

A man is standing on a beach holding a **snapper**.

A table with a **cauldron** in the dark.

A woman is posing for a picture with a **chiffon** dress.

A close up of a plate of food with a **scone**.

A **dumpling** sitting on top of a wooden table

A **saucepan** and a pot of food on a stove top.

A large **colander** with a piece of food on it.

A man holding a **banjo** in a park.

A large **chime** hanging on a metal pole

A **snowplow** truck driving down a snowy road.

A group of people standing around a large white **warship**.

A **okapi** is in the grass with a **okapi**.

A small brown and white **jackal** is standing in a field.

A large metal **candelabra** next to a wall.

A black and white photo of a **corkscrew** and a **corkscrew**.

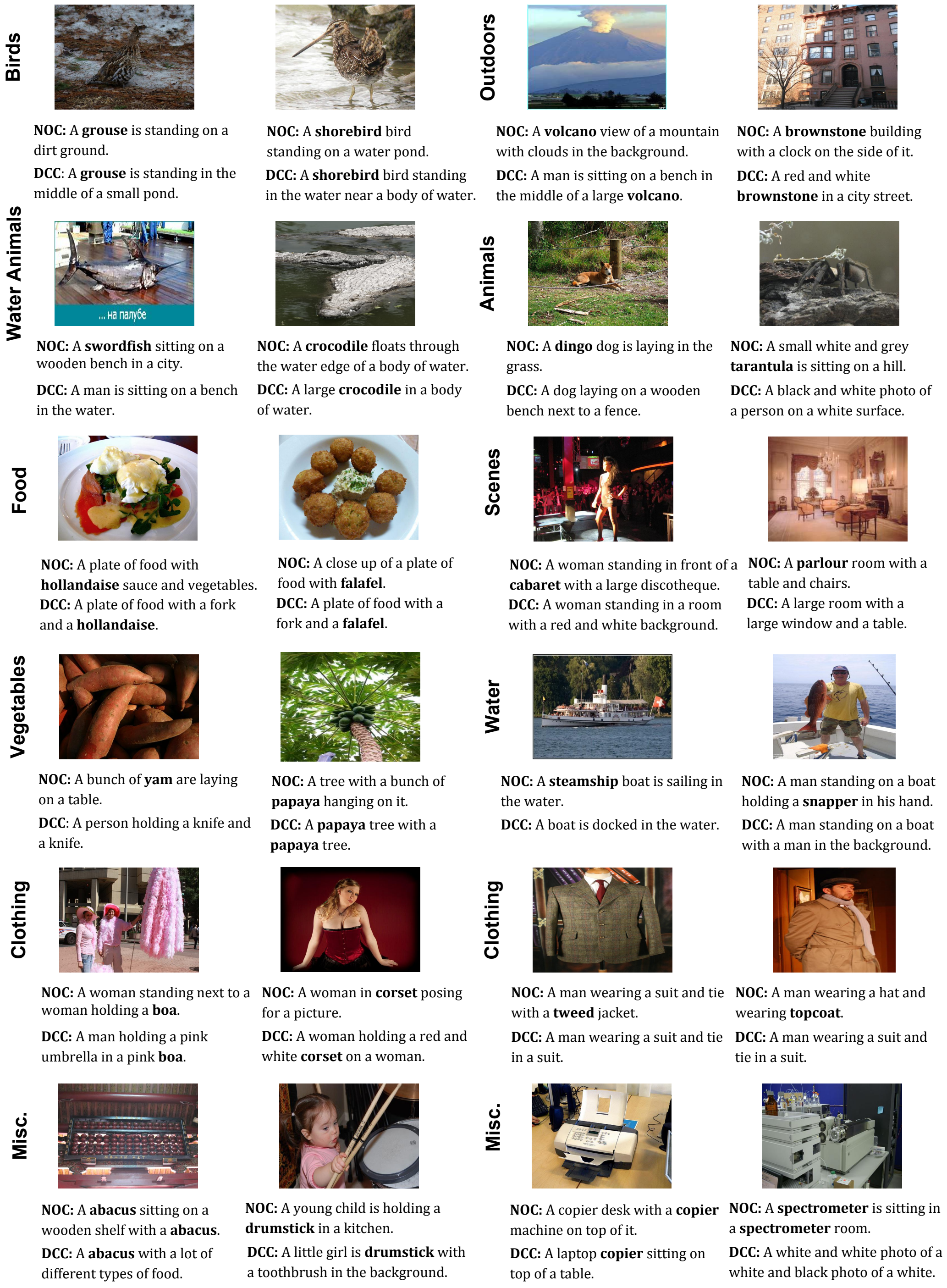
A **chainsaw** is sitting on a **chainsaw** near a **chainsaw**.

A man is sitting on a bike in front of a **waggon**.

A **volcano** view of a **volcano** in the sun.

A **trampoline** with a **trampoline** in the middle of it.

Figure 6.8: Examples of sentences generated by our NOC model on ImageNet images of objects belonging to a diverse variety of categories including food, instruments, outdoor scenes, household equipment, and vehicles. The novel objects are in **bold**. The last row highlights common errors where the model tends to repeat itself or hallucinate objects not present in the image.



NOC: A **grouse** is standing on a dirt ground.
DCC: A **grouse** is standing in the middle of a small pond.

NOC: A **shorebird** bird standing on a water pond.
DCC: A **shorebird** bird standing in the water near a body of water.

NOC: A **volcano** view of a mountain with clouds in the background.
DCC: A man is sitting on a bench in the middle of a large **volcano**.

NOC: A **brownstone** building with a clock on the side of it.
DCC: A red and white **brownstone** in a city street.

NOC: A **swordfish** sitting on a wooden bench in a city.
DCC: A man is sitting on a bench in the water.

NOC: A **crocodile** floats through the water edge of a body of water.
DCC: A large **crocodile** in a body of water.

NOC: A **dingo** dog is laying in the grass.
DCC: A dog laying on a wooden bench next to a fence.

NOC: A small white and grey **tarantula** is sitting on a hill.
DCC: A black and white photo of a person on a white surface.

NOC: A plate of food with **hollandaise** sauce and vegetables.
DCC: A plate of food with a fork and a **hollandaise**.

NOC: A close up of a plate of food with **falafel**.
DCC: A plate of food with a fork and a **falafel**.

NOC: A woman standing in front of a **cabaret** with a large discotheque.
DCC: A woman standing in a room with a red and white background.

NOC: A **parlour** room with a table and chairs.
DCC: A large room with a large window and a table.

NOC: A bunch of **yam** are laying on a table.
DCC: A person holding a knife and a knife.

NOC: A tree with a bunch of **papaya** hanging on it.
DCC: A **papaya** tree with a **papaya** tree.

NOC: A **steamship** boat is sailing in the water.
DCC: A boat is docked in the water.

NOC: A man standing on a boat holding a **snapper** in his hand.
DCC: A man standing on a boat with a man in the background.

NOC: A woman standing next to a woman holding a **boa**.
DCC: A man holding a pink umbrella in a pink **boa**.

NOC: A woman in **corset** posing for a picture.
DCC: A woman holding a red and white **corset** on a woman.

NOC: A man wearing a suit and tie with a **tweed** jacket.
DCC: A man wearing a suit and tie in a suit.

NOC: A man wearing a hat and wearing **topcoat**.
DCC: A man wearing a suit and tie in a suit.

NOC: A **abacus** sitting on a wooden shelf with a **abacus**.
DCC: A **abacus** with a lot of different types of food.

NOC: A young child is holding a **drumstick** in a kitchen.
DCC: A little girl is **drumstick** with a toothbrush in the background.

NOC: A copier desk with a **copier** machine on top of it.
DCC: A laptop **copier** sitting on top of a table.

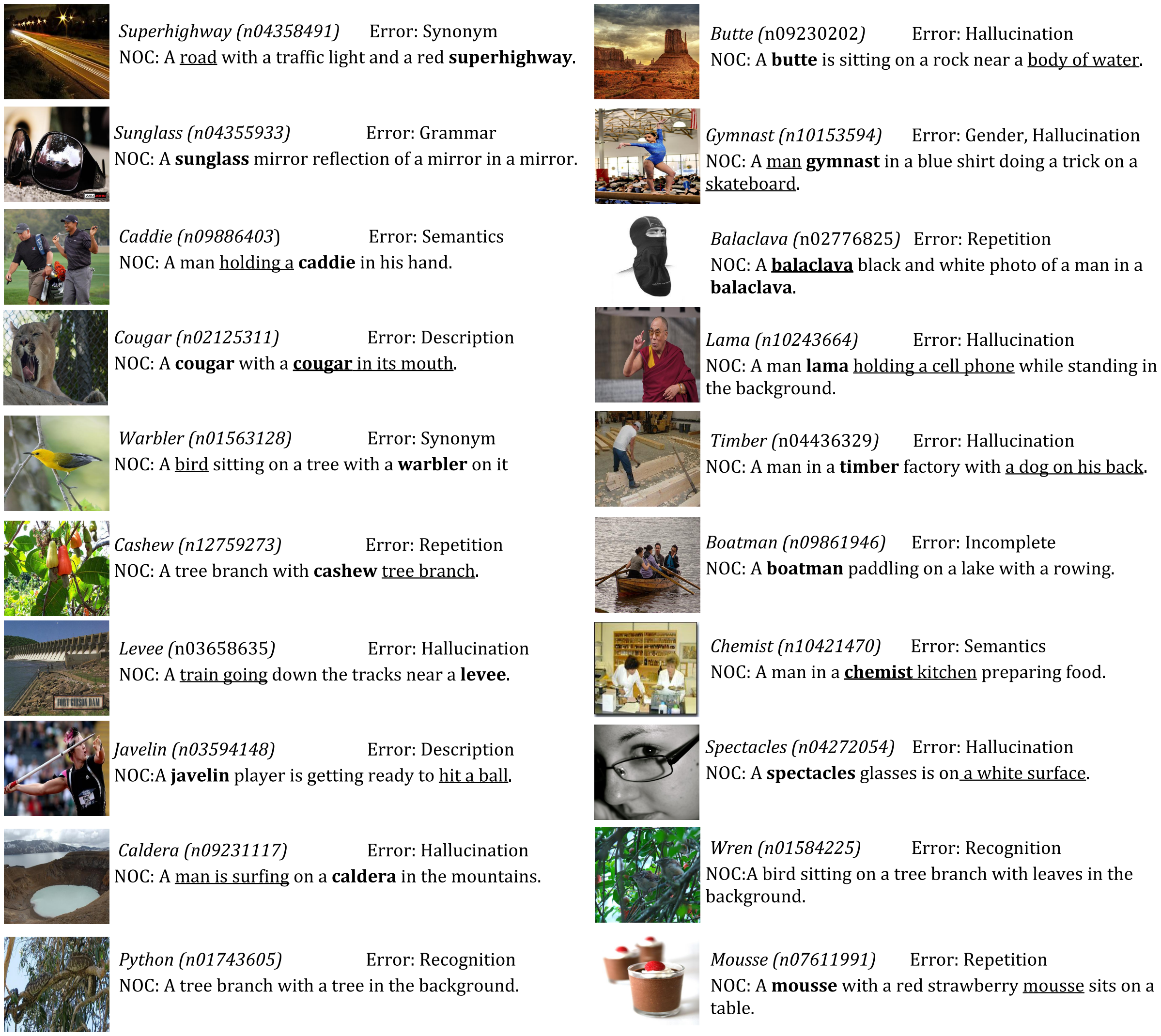
NOC: A **spectrometer** is sitting in a **spectrometer** room.
DCC: A white and white photo of a white and black photo of a white.

Figure 6.9: Examples comparing sentences generated by DCC (Hendricks *et al.*, 2016) and our NOC model on ImageNet images of object categories that *both* models can describe including food, animals, vegetables/fruits, indoor and outdoor scenes, and clothing. The novel objects are in **bold**.

6.5.3 Categorizing Errors

Fig. 6.10 presents some of the errors that our model makes when captioning Imagenet images. While NOC improves upon existing methods to describe a variety of object categories, it still makes a lot of errors. The most common error is when it simply fails to recognize the object in the image (e.g. image with 'python') or describes it with a more generic hyponym word (e.g. describing a bird species such as 'wren' or 'warbler' in Fig. 6.10 as just 'bird'). For objects that the model is able to recognize, the most common errors are when the model tends to repeat words or phrases (e.g. descriptions of images with 'balaclava', 'mousse' and 'cashew'), or just hallucinate other objects in the context that may not be present in the image (e.g. images with 'butte', 'caldera', 'lama', 'timber'). Sometimes, the model does get confused between images of other similar looking objects (e.g. it confuses 'levee' with 'train'). Apart from these the model does make mistakes when identifying gender of people (e.g. 'gymnast'), or just fails to create a coherent correct description even when it identifies the object and the context (e.g. images of 'sunglass' and 'cougar').

Relevant but Minor Errors. Fig. 6.11 presents more examples where NOC generates very relevant descriptions but makes some minor errors with respect to counting (e.g. images of 'vulture' and 'aardvark'), age (e.g. refers to boy wearing 'snorkel' as 'man'), confusing the main object category (e.g. 'macaque' with 'bear' and person as 'teddy bear') or makes minor word repetitions, and grammatical errors.



Superhighway (n04358491) Error: Synonym
NOC: A road with a traffic light and a red **superhighway**.

Sunglass (n04355933) Error: Grammar
NOC: A **sunglass** mirror reflection of a mirror in a mirror.

Caddie (n09886403) Error: Semantics
NOC: A man holding a **caddie** in his hand.

Cougar (n02125311) Error: Description
NOC: A **cougar** with a **cougar** in its mouth.

Warbler (n01563128) Error: Synonym
NOC: A bird sitting on a tree with a **warbler** on it

Cashew (n12759273) Error: Repetition
NOC: A tree branch with **cashew** tree branch.

*Levee (*n03658635*)* Error: Hallucination
NOC: A train going down the tracks near a **levee**.

Javelin (n03594148) Error: Description
NOC:A **javelin** player is getting ready to hit a ball.

Caldera (n09231117) Error: Hallucination
NOC: A man is surfing on a **caldera** in the mountains.

Python (n01743605) Error: Recognition
NOC: A tree branch with a tree in the background.

*Butte (*n09230202*)* Error: Hallucination
NOC: A **butte** is sitting on a rock near a body of water.

Gymnast (n10153594) Error: Gender, Hallucination
NOC: A man **gymnast** in a blue shirt doing a trick on a skateboard.

*Balaclava (*n02776825*)* Error: Repetition
NOC: A **balaclava** black and white photo of a man in a **balaclava**.

Lama (n10243664) Error: Hallucination
NOC: A man **lama** holding a cell phone while standing in the background.

*Timber (*n04436329*)* Error: Hallucination
NOC: A man in a **timber** factory with a dog on his back.

Boatman (n09861946) Error: Incomplete
NOC: A **boatman** paddling on a lake with a rowing.

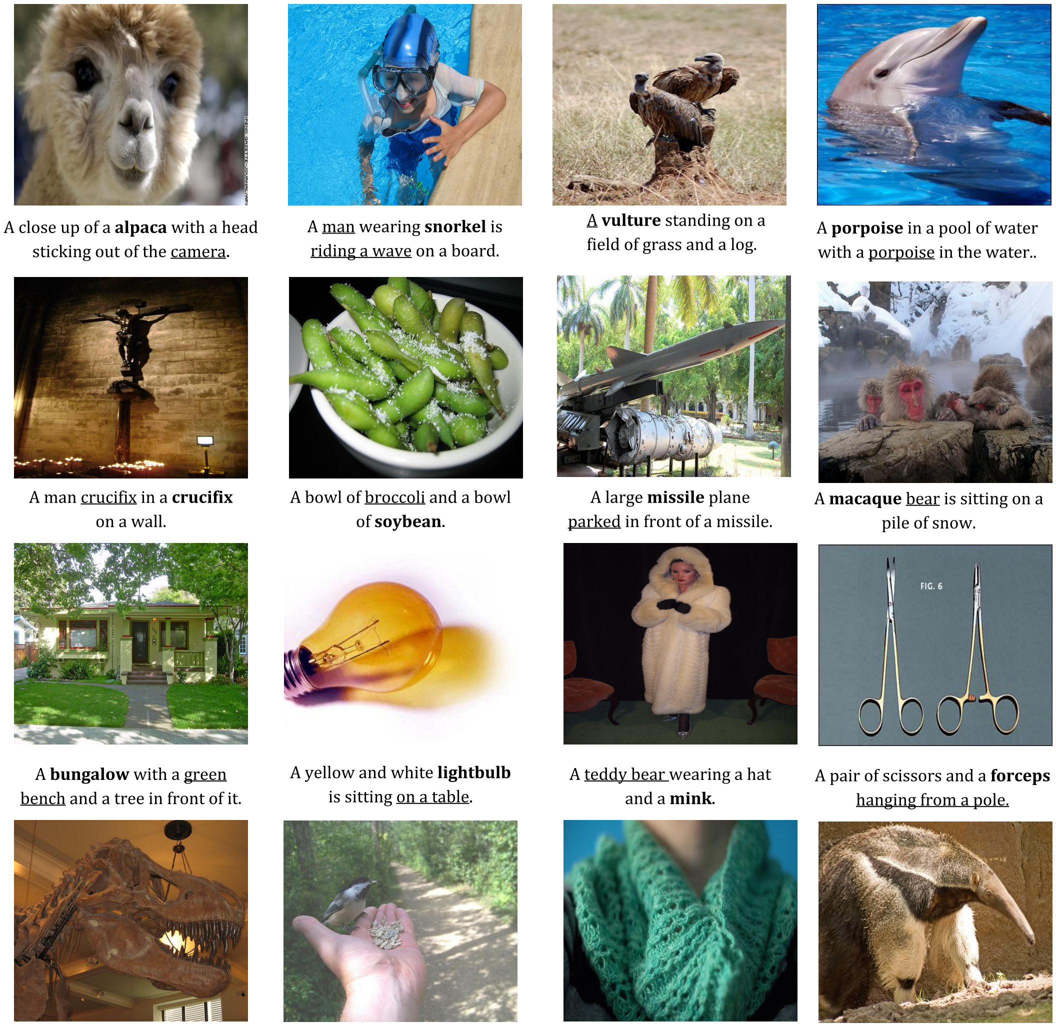
Chemist (n10421470) Error: Semantics
NOC: A man in a **chemist** kitchen preparing food.

Spectacles (n04272054) Error: Hallucination
NOC: A **spectacles** glasses is on a white surface.

Wren (n01584225) Error: Recognition
NOC:A bird sitting on a tree branch with leaves in the background.

Mousse (n07611991) Error: Repetition
NOC: A **mousse** with a red strawberry mousse sits on a table.

Figure 6.10: Examples of images where the model makes errors when generating descriptions. The novel object is in **bold** and the errors are underlined. NOC often tends to repeat words in its description, or hallucinate objects not present in the image. The model sometime misidentifies gender, misrepresents the semantics of the novel object, or just makes grammatical errors when composing the sentence.



A close up of a **alpaca** with a head sticking out of the camera.

A man wearing **snorkel** is riding a wave on a board.

A **vulture** standing on a field of grass and a log.

A **porpoise** in a pool of water with a porpoise in the water..

A man crucifix in a **crucifix** on a wall.

A bowl of broccoli and a bowl of **soybean**.

A large **missile** plane parked in front of a missile.

A **macaque** bear is sitting on a pile of snow.

A **bungalow** with a green bench and a tree in front of it.

A yellow and white **lightbulb** is sitting on a table.

A teddy bear wearing a hat and a **mink**.

A pair of scissors and a **forceps** hanging from a pole.

A **tyrannosaurus** statue of a tyrannosaurus statue in a museum.

A **chickadee** bird sitting on a bird food near a bird.

A close up of a person wearing a bolero and a **cashmere**.

A couple of **aardvark** standing next to a large rock wall.

Figure 6.11: Some examples where NOC makes minor errors when describing the image. The novel object is in **bold** and the word or segment corresponding to the error is underlined. Counting, repetitions, confusing object categories (e.g. 'macaque', 'bear'), grammatical errors, and hallucinating objects that are absent are some common errors that the model makes. However, the generated description is still meaningful and relevant.

6.6 Conclusion

We presented an end-to-end trainable architecture that used knowledge from external sources to generate descriptions for object classes unseen in paired image-caption data. Notably, NOC's architecture and training strategy enables the visual recognition network to retain its ability to recognize several hundred categories of objects even as it learns to generate captions on a different set of images and objects. We demonstrate our model's captioning capabilities on a held-out set of MSCOCO objects as well as several hundred ImageNet objects. Both human evaluations and quantitative assessments show that our model is able to describe many more novel objects compared to previous work while also maintaining or improving descriptive quality.

Code and generated sentences for this model along with additional examples and details about the human evaluations are available at the following link: https://vsubhashini.github.io/noc.html

Chapter 7

Segmenting and Describing Longer Videos

This chapter introduces the task of temporal segmentation and description, where a vision system needs to localize and caption events worth describing in movie-length videos. As seen in Chapter 4, existing video captioning systems focus on describing a single event in a very short video segment; in contrast, the temporal segmentation and description task requires identifying salient events in a long clip and describing them in context. To address the task, we propose a novel framework which processes a video generating foreground proposals comprising noteworthy segments and later refines those segments to generate textual descriptions. We combine a scene change predictor for temporal segmentation with an architecture that uses both a convolutional network and a bi-directional recurrent neural network encoder to select foreground proposals. Foreground features from the encoder are then provided as an input to a recurrent language model decoder to generate descriptions. We present and evaluate our network on two large movie datasets which consist of over 350 hours of video from 186 movies with over 120,000 segments.

7.1 Temporal Segmentation Networks for Localization and Description of Videos

As mentioned previously, the ability to identify events in long videos and describe them is an important step towards video understanding. For most humans recognizing key events in videos, and describing the narrative to varying degrees of detail comes easily. However, this remains extremely challenging for modern visual recognition systems. Research in computer vision has made significant advances on different aspects of the problem, foremost among these are recognizing objects in images (Krizhevsky *et al.*, 2012, He *et al.*, 2016), and categorizing activities in videos (Gorban *et al.*, 2015). In this thesis we have seen several research works on the expressive task of describing visual content in natural language. Chapter 2 discussed some of the recent progresses in image captioning

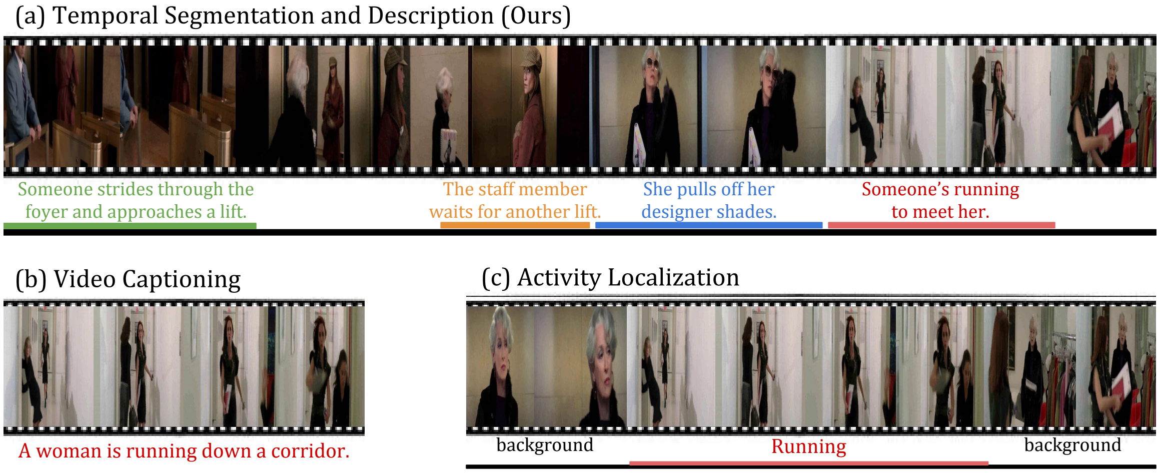

(a) Temporal Segmentation and Description (Ours)
Someone strides through the foyer and approaches a lift.
The staff member waits for another lift.
She pulls off her designer shades.
Someone's running to meet her.
(b) Video Captioning
A woman is running down a corridor.
(c) Activity Localization
background
Running
background

Figure 7.1: (a) We address the task of temporal segmentation and description of movies. This task extends both (c) event localization and (b) video captioning, performing them jointly for longer movie clips.

(Donahue *et al.*, 2015, Vinyals *et al.*, 2015), and Chapters 3 to 5 presented progress in video captioning (Venugopalan *et al.*, 2015b;a; 2016).

However, despite the growing interest in video captioning, recent research including those discussed in Chapters 3 to 5 have focused predominantly on describing single events or activities in short clips. Progress has been made in improving the visual representation to better capture temporal information during captioning (Venugopalan *et al.*, 2015a, Yao *et al.*, 2015, Pan *et al.*, 2016), but these still target single events. Orthogonally, there has also been work on activity localization, which is the task of recognizing and identifying the boundaries of a single activity in an untrimmed video (Shou *et al.*, 2016, Singh *et al.*, 2016, Escorcia *et al.*, 2016, Yeung *et al.*, 2016, Ma *et al.*, 2016). But progress in this area has lagged considerably, compared to the analogous task of object detection in images, due to scarcity of annotated video data. The next key research challenge that we address in this work is joint detection and captioning of activities. In real world applications such as human-robot interaction, surveillance, or describing movies for the blind, there is a long, continuous stream of video from which one needs to identify segments or salient events before describing them. Hence, it is important to develop models that can both identify and localize interesting events in long videos, and generate descriptions for them.

In this work, we take a step towards this direction. We introduce the joint temporal-segmentation and description task motivated by generating descriptions of movies for the visually impaired. As depicted in Fig.7.1, the goal is to predict a set of descriptions across segments of a clip from a movie for the purpose of movie description. However, defining "events and activities worth describing" in generic videos can be quite challenging in itself. This thesis defines "events worth describing" in the context of generating DVS (descriptive video service), a separate audio track for the visually impaired describing the visual elements on screen. In video captioning DVS can be viewed as a text modality that our model needs to output[1]. By tying our definition of "events" to DVS descriptions, the eventual goal of the model is to take context into account in order to generate useful descriptions for the visually impaired.

For this task, we utilize and adapt the annotations from the two large movie description corpora seen in Chapters 4 and 5, namely the MPII Movie Description (MPII-MD) dataset (Rohrbach *et al.*, 2015b) and Montreal Video Annotation Dataset (MVAD) (Torabi *et al.*, 2015). More precisely, we use the annotations from the Large Sacle Movie Description Challenge (LSMDC 2016) (Rohrbach *et al.*, 2017) which are manually aligned to the exact short segment of the movie clip where the described event appears. As opposed to the data in MPII-MD and MVAD, which only provide short 4-6 second snippets corresponding to DVS segments in the movies, our adapted datasets consist of minute-long clips encompassing almost the complete movie along with annotations for the segment boundaries, as well as the manually aligned DVS descriptions from the original datasets. Our clips cover over 325 hrs of video from 186 movies, and contain 124,806 segments with DVS descriptions.

To address this task, we propose a novel Temporal Segmentation and Description Network (TSDN) that aims to jointly segment a long video in time and describe segments of interest. Our model architecture, as seen in the previous chapters, consists of both a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN), i.e. CNN-LSTM based models in video (Chapter 3)

[1] The task of generating the actual audio track itself is out of scope of this thesis since it involves several more components including interleaving of the generated description into the existing audio track at appropriate locations.

and image captioning (Vinyals *et al.*, 2015, Donahue *et al.*, 2015, Johnson *et al.*, 2016). We build on the S2VT encoder-decoder architecture from Chapter 4 by incorporating a bi-directional LSTM encoder as well as a novel temporal segmentation layer in order to predict salient DVS segments. As we report below, our TSDN method performs considerably better than naive baselines which tile the video into contiguous clips or segments using conventional change detection algorithms, and describe each segment.

7.2 Temporal Segmentation Description Network

Overview. Our goal is to jointly segment multiple salient events in a video while also describing them. As with previous video captioning work, we employ a CNN-LSTM style model; however the key distinguishing factor in our model is being able to generate descriptions for multiple segments and not just a single sentence describing the complete input. More importantly, we want to first temporally segment the video to identify coherent events. Additionally, we want the captioning module to have a local context from the segmentation module as well as a global context of the event within the clip. We address these constraints by drawing elements from recent works in video summarization (Potapov *et al.*, 2014, Zhang *et al.*, 2016, Baraldi *et al.*, 2016), and video captioning (Venugopalan *et al.*, 2015a, Yao *et al.*, 2015).

We first use an unsupervised technique to find change points indicating coherent video segments. We then learn a supervised bi-directional LSTM model to predict foreground and background segments. Finally, the features from the bi-directional LSTM are used with an LSTM language model to generate descriptions. The full architecture of our model is described in the following subsections and is also illustrated in Figure 7.2.

7.2.1 Convolutional Network

To model visual inputs, we use the VGG-16 architecture (Simonyan and Zisserman, 2014b) as it has strong object-recognition performance. This also helps make a consistent comparison with the S2VT model discussed in Chapter 4. In particular, as seen in prior work, we use the activations from the penultimate layer

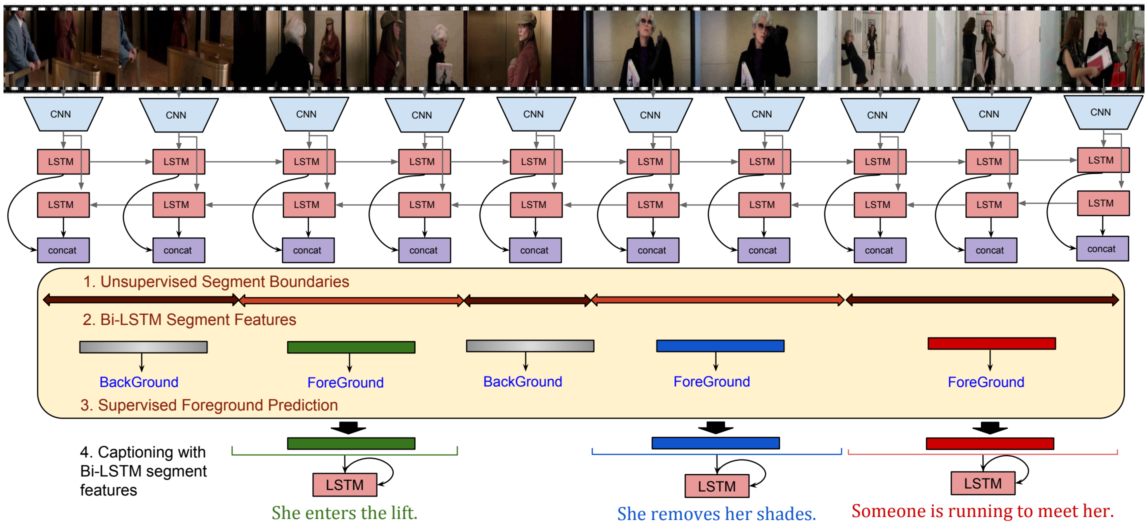

CNN
LSTM
LSTM
concat
1. Unsupervised Segment Boundaries
2. Bi-LSTM Segment Features
BackGround
ForeGround
BackGround
ForeGround
ForeGround
3. Supervised Foreground Prediction
4. Captioning with Bi-LSTM segment features
LSTM
She enters the lift.
She removes her shades.
Someone is running to meet her.

Figure 7.2: Overview: We use an unsupervised segmentation algorithm to first obtain non-overlapping intervals in the video. We then use a CNN and a bi-directional LSTM encoder to process frames in the video. The outputs from the forward and backward LSTMs are combined to generate segment features for each of the non-overlapping intervals. The model is trained to predict foreground and background segments from the encoder's segment features and then generate a description for each foreground segment using an LSTM decoder.

(with reLu applied after the fully-connected layer before classification) while keeping the rest of the network fixed. This provides a 4096 dimension vector which is then embedded to a lower 512 dimension vector with another fully connected layer and provided as input to the Bi-LSTM encoder.

7.2.2 Bi-directional Temporal Processing

We use the sequence-to-sequence encoder-decoder framework seen in S2VT and other recent end-to-end video description networks (Yao *et al.*, 2015, Pan *et al.*, 2016, Yu *et al.*, 2016). However, unlike previous methods that process frames in a single sequential order using recurrent neural networks to encode the video, we use a bi-directional LSTM network (Schuster and Paliwal, 1997) for modeling stronger sequential information from both the past and the future. Modeling the video from both directions is particularly relevant in our case, since we not only want a representation of the video for description, but we also need information to temporally select segment belonging to foreground events. The bi-directional LSTM helps us achieve this.

Recall from Equation (2.7), the set of recurrences in the LSTM encoder, and Equation (3.2) which represents the log probability of the generated sentence. Essentially, in the encoding phase, given an input sequence X $(x_1, \ldots, x_n)$, the LSTM computes a sequence of hidden states $(h_1, \ldots, h_n)$. During decoding it defines a distribution over the output sequence Y $(y_1, \ldots, y_m)$ given the input sequence X as $p(Y|X)$ is

$$p(y_1, \ldots, y_m | x_1, \ldots, x_n) = \prod_{t=1}^{m} p(y_t | h_{n+t-1}, y_{t-1}) \tag{7.1}$$

where the distribution of $p(y_t|h_{n+t})$ is given by a *softmax* over all of the words in the vocabulary. In LSTMs seen in earlier chapters, the first hidden state of the decoder, h_{n+t}, is obtained from the last state and prediction of the encoder, h_{n+t-1}, y_{t-1}, based on the recursion in Equation (2.7). In a bi-directional LSTM (Bi-LSTM), the encoder computes two sets of recurrent state variables, one in the forward direction, and one in the backward direction. This model is depicted in Fig. 7.2; note that the forward and the backward chains do not directly interact. We combine the information in those two chains by concatenating them and embedding them.

$$h_t = f(h_t^{forward}, h_t^{backward})$$

The combined state thus contains information about both past and future events in the sequence.

7.2.3 Temporal Segmentation

For temporal segmentation, we use both an unsupervised change detection algorithm to obtain segment boundaries as well as a supervised bi-directional LSTM to predict foreground segments. To predict segments in a supervised manner, we extract the output activations from the Bi-LSTM for each segment and learn to predict foreground and background content based on the presence of a description for the event segment. The presence of DVS description provides supervision for foreground prediction. We first explain the unsupervised temporal segmentation method and then the supervised fore ground prediction approach.

We first employ an unsupervised kernel-based change point detection algorithm Kernel Temporal Segmentation (KTS) (Potapov *et al.*, 2014) to identify changes in the underlying visual signal (sequence of visual features in our case). While simple shot-boundary detection algorithms focus primarily on video transitions, in order to effectively identify relevant foreground events and in particular signal change-points our model needs to look at the full signal across a longer sequence of frames. We accomplish this by using a bi-directional LSTM in conjunction with the Kernel Temporal Segmentation algorithm.

To identify signal change-points with KTS, similar to our prior CNN-LSTM models, we start with a CNN's (specifically VGG16's (Simonyan and Zisserman, 2014b)) fully connected layer descriptors/features to form the frame similarity matrix or positive definite kernel. Stating this more precisely, for the sequence of video descriptors $x_i \in X, i = 0, \ldots, n-1$, let $K : X \times X \to R$ be a kernel function between descriptors. Let $\mathcal{H}$ be the feature space of the kernel $K(\cdot,\cdot)$. Denote $\phi : X \to \mathcal{H}$ the associated feature map, and $\| \cdot \|_{\mathcal{H}}$ the norm in the feature space $\mathcal{H}$. We use a slight modification of KTS that minimizes the following objective,

$$\underset{m;t_0,\ldots,t_{m-1}}{\text{Min}} \sum_{i=0}^{m} v_{t_{i-1},t_i} \tag{7.2}$$

where v_{t_{i-1},t_i} is the within-segment kernel variance:

$$v_{t_i,t_{i+i}} = \sum_{t=t_i}^{t_{i+1}-1} \| \phi(x_t) - \mu_i \|_{\mathcal{H}}^2 \tag{7.3}$$

$$\mu_i = \frac{\Sigma_{t=t_i}^{t_{i+1}-1} \phi(x_t)}{t_{i+1} - t_i} \tag{7.4}$$

and μ_i denotes the segment mean.

Algorithm. First the kernel is computed for each pair of descriptors in the sequence, here it's the Gram matrix or inner product of all pairs of descriptors. Then for each possible starting point t and segment duration d, the segment variances are computed. This can be done efficiently by precomputing the cumulative sums of the matrix (Potapov *et al.*, 2014). Then the objective in Equation (7.2) is minimized using a dynamic programming (DP) algorithm. The DP algorithm iteratively computes the best objective value for the first j descriptors and i change points. Finally, the optimal segmentation is reconstructed by backtracking. This generates non-overlapping disjoint segments of the video clip.

7.2.4 Supervised Foregound Selection

Segment Features. For each of the segments obtained from KTS, based on the predicted start and end, we select the output of the bidirectional LSTM on the full clip at both the beginning and end of the segment, as well as the mean-pooled descriptor as features for the proposed segment. This generates a fixed length feature vector for each segment. These fixed length feature vectors form the input for both the segment foreground prediction layer as well as the input to an LSTM sequence model decoder for description as described next.

Foreground prediction. We add a fully-connected layer and a binary classification softmax layer to predict each unsupervised segment as either foreground or background. The groundtruth data for supervision is obtained based on overlap and presence of DVS annotation. Specifically, we identify all KTS segments which have an Intersection Over Union (IOU) of greater than 0.5 with any groundtruth DVS segments and treat them as positive, and all segments with IOU less than 0.2 as negative. As seen in object detection models (Girshick *et al.*, 2014, Ren *et al.*, 2015b), segments with IOU value between 0.2 and 0.5 are ignored. The bidirectional LSTM takes the sequence of frame features, the unsupervised segment end-points, and the segment labels as input, generating a batch of segment features and learns to predict each segment as either foreground are background.

7.2.5 LSTM decoder for Description

Our final description model follows previous video captioning encoder-decoder works (Chapters 3 to 5) by using an LSTM recurrent neural network for the language model. For language generation, we use a forward uni-directional LSTM network. First, the word tokens in the captions are embedded to a dense representation to produce vectors $(x_1, \dots, x_m)$. Then the LSTM model is given the segment features as input i.e. the output from the Bi-LSTM encoder embedded to a lower dimension, then the mean-pooled segment descriptors embedded to a lower dimension, as well as a special beginning of sentence <BOS> token in that order. The LSTM computes a sequence of hidden states (h_t) and output vectors (y_t) for each word/item in the input sequence using the recurrence $h_t, y_t = f_{LSTM}(h_{t-1}, x_t)$, where f_{LSTM} refers to the LSTM recurrence in Equation 2.7. If the LSTM model is parametrized by θ, the model is trained to predict the caption using the cross-entropy loss to optimize the following log-likelihood equation (similar to Equation (7.5))

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{m} \log p(y_t | x_{seg}, y_{t-1}; \theta) \tag{7.5}$$

x_{seg} refers to the features of the segment, and for simplicity we overload the output y_t to be outputs after the softmax function is applied to the LSTM output. The model predicts one word at a time until it outputs the end-of-sequence (EOS) token.

We note here that for language generation, we train an independent bidirectional LSTM encoder and decoder models i.e., the segment foreground prediction bi-directional LSTM encoder does not share any of it's parameters with the encoder used for segment description

7.2.6 Training and Optimization

We embed activations from the convolutional network to a lower 512 dimensional vector. Our LSTMs, both the bi-directional encoders as well as the language decoder use hidden layers of 512 dimensions. Our bi-directional LSTM encoders were unrolled to a maximum of 150 steps, and the captioning LSTM decoder was unrolled to 20 steps. For KTS, we chose the expected number of change points to be 12 based on the average number of segments per clip on both datasets. For segment foreground prediction, we trained 32 clips per batch where each clip contained 12 segments. Each batch had roughly 40% positive segments and remaining negative. In order to train the captioning model on segments, we used a batch size of 16 clips per batch (training for a single segment per clip). The clips themselves were pre-processed at the rate of 2 frames per second to generate about

120-150 frames per clip. For clips that had fewer than 150 frames, the LSTM inputs were padded with 0s. We use ADAM with a learning rate 1e-4 for optimization.

7.3 Datasets

We make available two large movie description datasets for the task. Specifically, we expand on existing MPII-MD (Rohrbach *et al.*, 2015b) and MVAD (Torabi *et al.*, 2015) movie description datasets and adapt them for the task of segmentation and captioning. MPII-MD and MVAD datasets contain short 4-6s segments from 94 and 92 movies respectively. Each segment is annotated with a one or two sentence DVS description explicitly describing what appears on the screen. In this work, our goal is to segment and describe longer videos and not just single segments. Hence we look at the original full length movies used to create the MPII-MD and MVAD datasets, and obtain the time-stamped DVS annotations from the LSMDC16 (Rohrbach *et al.*, 2017) challenge.

We pre-process the full length movies and cut them into roughly 1 minute clips ensuring that DVS segments are not split across clips. This pre-processing helps in multiple ways, 1) it makes the bi-directional LSTM encoder computation reasonably sized (150 steps) during training, 2) it creates sufficient number of clips for training and evaluation of LSTM models (20,349 clips altogether), and 3) it ensures that most clips have a reasonable number of segments in each (about 6 segments on average). Collectively the clips have a total duration of over 325hrs, and there are roughly 6 DVS segments per clip (actual number of segments varies between 1 to at most 18). We retain the set of movies in the train, validation, and test splits of the original data sources. More detailed comparison of the data for temporal segmentation and description and the original datasets are provided in Table 7.1.

We note here that although we pre-process the movies to obtain shorter clips, this process is only necessary for training our models. At test time it is possible to use the proposed methods to generate descriptions for full movies e.g., KTS segmentation can be first applied to cut the test movies at longer 1-2 minute intervals before processing. However, this can make evaluation considerably challenging. Hence we pre-process test movies as well.

Our dataset	MPII-MD	MVAD
Number of Movies	94	92
Number of Clips	11,560	8,789
Avg. clip length	57s	58s
Total duration of clips	184h,46m	141h,42m
Avg. segments per clip	6	6
Segments - LSMDC16 challenge		
Number of segments	68,375	56,431
Total duration of segments	68hrs	97hrs
Avg. segment length	3.9s	6.2s

Table 7.1: Corpus Statistics of the two adapted movie datasets introduced in this work.

7.4 Experiments

Our model takes a single video clip as input and predicts temporal segments with corresponding confidence scores and captions. Our goal is to evaluate performance on the ability of the model to identify segments worth describing and generating the description. We first describe the evaluation metrics and then present the results of the models.

7.4.1 Evaluation Metrics

Since the temporal segmentation and description task involves both localization as well as captioning we evaluate on each of the tasks independently. Our metrics for evaluation are similar to the ones used for dense-captioning (Johnson *et al.*, 2016).

Localization To evaluate the segmentation localization ability alone for the methods, we use intersection over union (IOU) across different thresholds $t = \{0.4, 0.5, 0.6\}$. More concretely, we measure the IOU of the proposed segments against the ground-truth segments to cover all ground-truth segments i.e., for each groundtruth segment we identify a distinct predicted segment which has the highest overlap and consider the extent of overlap (IOU) when applying the threshold. E.g., if a groundtruth segment is covered best by a predicted segment with an IOU of 0.55, then at a $t = 0.5$ the prediction is considered positive, but at $t \geq 0.6$ it is considered negative. We then evaluate recall and precision at each value of the threshold, reporting $recall$ at t, and $precision$ at t for each value of t. We also report the F1 score at $t = 0.5$ (i.e., IOU=0.5 or greater).

Captioning For captioning, as seen in previous Chapters 4 and 5 for DVS we use the automated machine translation metrics described in Chapter 3: METEOR (Denkowski and Lavie, 2014), BLEU (Papineni *et al.*, 2002) and Rouge-L (Lin, 2004). We consider evaluating the generated captions in two ways: 1) without localization, and 2) with localization. To evaluate just the captioning performance alone (without taking localization into account), for each clip we combine the ground truth captions for all segments within the clip to create a bag of reference captions for the clip. We then evaluate the predicted captions against the ground-truth references to determine the METEOR scores for the predictions. We report the average METEOR scores across all predicted captions. This evaluation is based on the image captioning work in Johnson *et al.* (2016). To evaluate captions when considering localization, for each groundtruth segment we find the predicted segment with the highest overlap, and then measure the METEOR score of the predicted segment's caption against the single groundtruth reference caption. This allows us to evaluate the overall quality of the description more precisely.

We note that it is necessary to consider both of these metrics, F1 localization and METEOR captioning score, in conjunction when determining the performance of a model. Neither metric can individually capture a model's joint ability to localize and describe content.

7.4.2 Methods for Comparisons

We present and compare our approach against multiple segmentation and captioning approaches. We describe the individual temporal segmentation and captioning approaches before presenting our results.

Segmentation

We compare 5 different approaches to generate the temporal segments: scene-subshots, uniform segments, KTS segments, frame-level foreground and background predictions (FGBG), and the segments generated by our TSDN model.

Scene-Subshot. As an initial baseline, we use a scene subshot detection algorithm that examines the absolute differences in pixels (Richardson, 2004) between each consecutive pair of frames. We determine a frame as the beginning of a new segment if there is more than 40% change between two subsequent frames in the video. Scene subshot algorithms, are particularly good at capturing video transitions such as fade-ins and shot changes. So each of the subshots represent temporal coherence from the perspective of individual pixel differences. We consider all subshots to be foreground segments.

Uniform. We also compare against a baseline that segments the video at uniform intervals based on the average segment duration in the training datasets. For MPII-MD we select segments at every 4.5s interval, and for MVAD at every 8s interval. Similar to scene-subshots, we again consider all segments/intervals to be in the foreground hence these segments cover the entire clip/movie.

KTS. The next method we use is the Kernel Temporal Segmentation (KTS) model's change point detection (Potapov *et al.*, 2014). As described in Sec.7.2.3, KTS identifies change points corresponding to the boundaries of temporal segments based on the similarity between all pairs of frames. As in the previous cases we consider all segments to be in the foreground so as to cover as much of the video.

FGBG. In addition to the above methods, we present a bidirectional-LSTM based approach that predicts each frame as foreground and background content (referred to as FGBG). This approach is similar to the work by Zhang *et al.* (2016) for video summarization where they train a bi-directional LSTM to predict a confidence score for whether each frame belongs to the foreground. In our case we train a binary classifier based on the bi-LSTM output for each frame. During training, all

Models	Recall @0.4	 @0.5	 @0.6	Precision @0.4	 @0.5	 @0.6	F1 @0.5
Scene Subshot	11.9	08.9	05.8	13.9	10.1	06.6	09.5
KTS (Potapov *et al.*, 2014)	**54.6**	**42.9**	**26.8**	26.2	19.6	12.1	26.9
FGBG (Zhang *et al.*, 2016)	43.9	32.3	19.0	24.7	17.2	10.0	22.5
Uniform	42.5	26.2	13.6	27.6	16.8	08.7	20.5
TSDN (Ours)	46.2	37.7	24.4	**32.2**	**26.3**	**17.0**	**31.0**
TSDN Oracle	65.9	53.1	34.2	79.2	63.8	41.0	58.0

Table 7.2: Dataset: MPII-MD. Recall and Precision @ different thresholds of IOU for the predicted segments, and F1@IOU≥ 0.5. Values in percentage ($\%$), higher is better.

frames that correspond to a groundtruth DVS segment are considered positive and the rest negative.

TSDN TSDN denotes our own approach from Figure 7.1 that trains a bi-LSTM on KTS intervals to predict foreground segments.

TSDN Oracle. This denotes the upper bound on segmentation based on our TSDN approach. Recall that during training of the TSDN model, we select groundtruth foreground segments as those KTS segments with an IOU threshold of 0.5 or greater to be positive, and the those with lesser than IOU 0.2 as negative. The results for the oracle are based on considering all the positive segments as foreground. This provides an upper bound for the TSDN approach.

7.4.3 Results

Localization. We first compare different segmentation approaches on the MPII-MD dataset in Table 7.2 and MVAD dataset in Table 7.3. On the MPII-MD dataset, the KTS approach has good recall at different thresholds, however although TSDN has a lower recall it has a much higher precision and hence a significantly better F1 score. On the MVAD dataset however, just generating uniform segments gives a very high recall, and precision comparable to that of the KTS method. Our TSDN achieves a higher precision at all thresholds and hence a higher F1 overall. Viewing

Models	Recall @0.4	Recall @0.5	Recall @0.6	Precision @0.4	Precision @0.5	Precision @0.6	F1 @0.5
Scene Subshot	18.9	15.4	10.5	26.8	20.8	13.9	17.7
KTS (Potapov *et al.*, 2014)	59.4	49.5	33.6	37.1	29.6	19.8	36.7
FGBG (Zhang *et al.*, 2016)	47.2	36.2	21.7	34.1	24.8	14.6	29.5
Uniform	**75.6**	**60.4**	**39.0**	36.1	28.9	18.6	39.1
TSDN (Ours)	40.2	34.7	24.6	**56.9**	**49.2**	**34.8**	**40.7**
TSDN Oracle	70.8	60.0	41.7	84.2	71.4	49.6	65.2

Table 7.3: Dataset: M-VAD. Recall and Precision @ different thresholds of IOU for the predicted segments, and F1@IOU≥ 0.5. Values in percentage ($\%$), higher is better.

the segmentation methods collectively, scene subshot has the worst performance, hence simple scene change indication is not a good segmentation approach. FGBG method that makes frame-wise foreground and background prediction has a worse performance compared to KTS. While KTS performs consistently on both datasets, Uniform segmentation does quite poorly on the MPII-MD dataset but surprisingly well on the M-VAD dataset. Our TSDN approach has higher precision and F1 compared to all other segmentation approaches.

Captioning Tables 7.4 and 7.5 present automated caption scores (without taking localization into account) of the S2VT and Bi-LSTM captioners on the generated segments. Captioning scores for the segments do not appear to vary quite as much as the localization scores for a particular captioning method. However, it is easy to observe that the Bi-LSTM model does considerably better than S2VT on METEOR on both datasets, as well as on other metrics on M-VAD, but scores on other metrics for S2VT on the MPII-MD datasets are higher. As noted before, the evaluations presented in Tables 7.4 and 7.5 overcome the drawback of comparing the generated sentences against a single groundtruth reference, but they fail to take into account the effect of localization.

To evaluate the caption quality taking localization into account we evaluate the captions by selecting the predicted segment closest to the groundtruth DVS segment. Specifically, for each groundtruth segment, we identify the nearest pre-

	S2VT				Bi-LSTM			
Models	B-2	B-1	Rouge	METEOR	B-2	B-1	Rouge	METEOR
Scene Subshot	7.2	29.6	23.5	11.7	6.0	25.1	21.3	14.1
KTS [2]	7.6	32.8	24.7	11.6	6.4	26.9	22.1	14.0
FGBG [3]	7.6	33.3	24.7	11.8	6.8	28.2	22.5	14.0
Uniform	7.3	32.1	24.7	12.1	6.7	27.2	22.3	14.2
TSDN (Ours)	7.7	33.5	25.0	11.7	6.3	27.1	22.1	14.1
Groundtruth	10.1	38.6	27.4	12.7	8.4	31.7	24.6	15.1

Table 7.4: Dataset: MPII-MD. Caption evaluation (without localization). BLEU@2 (B-2), BLEU@1 (B-1), ROUGE-L (Rouge) and METEOR scores for generated captions. Values in percentage (%), higher is better.

dicted segment's caption and compare with the single groundtruth reference for that segment. The results are present in Table 7.6. Since we compare against just a single reference, we use only the METEOR metric which is more robust when there are fewer reference (Section 4.4.2). The scores here are lower than the ones in Tables 7.2 and 7.3 because of comparing against just a single reference. As before, the captioning performance does not appear to differ much across the different segmentation approaches. One reason for this is that aside from our TSDN approach all other methods generate contiguous segments, hence there is always some predicted segment that overlaps with the groundtruth segment. The results also show that the BiLSTM approach considerably outperforms S2VT in captioning on the METEOR metric.

7.5 Qualitative examples

We present a few qualitative examples in Figures 7.3 and 7.4. Each example shows a few video frames from a portion of the clip. Below the frames we present the foreground segmentations generated by our TSDN approach, along with the groundtruth, uniform and KTS segmentations. We also show the descriptions generated by our model and the groundtruth descriptions.

[2] Potapov *et al.* (2014)

[3] Zhang *et al.* (2016)

	S2VT				Bi-LSTM			
Models	B-2	B-1	Rouge	METEOR	B-2	B-1	Rouge	METEOR
Scene Subshot	3.8	22.1	17.0	10.5	5.9	30.5	25.0	15.0
KTS [4]	3.0	24.5	17.7	9.1	6.1	31.4	25.0	15.0
FGBG [5]	3.7	25.2	18.4	10.0	7.0	32.8	25.8	15.2
Uniform	2.5	23.7	15.7	9.3	6.5	32.1	25.3	15.2
TSDN (Ours)	3.3	27.2	18.5	9.2	7.1	33.7	26.1	15.3
Groundtruth	2.9	26.4	17.9	9.1	8.0	35.7	27.3	16.1

Table 7.5: Dataset: M-VAD. Caption evaluation (without localization). BLEU@2 (B-2), BLEU@1 (B-1), ROUGE-L (Rouge) and METEOR scores for generated captions. Values in percentage (%), higher is better.

Dataset	Model	Scene Subshot	KTS	FGBG	Uniform	TSDN
MPII-MD	S2VT	6.9	6.6	6.5	6.8	6.5
	BiLSTM	8.8	**8.9**	8.8	**8.9**	**8.9**
MVAD	S2VT	6.9	6.9	6.7	6.9	7.6
	BiLSTM	8.5	**8.7**	8.6	**8.7**	**8.7**

Table 7.6: Caption evaluation (with localization): METEOR scores (in %). For each groundtruth segment we identify the predicted segment with the highest overlap and evaluate the generated caption against the single groundtruth reference. Since scene-subshot, KTS, and uniform segmentation proposals do not have any background segments, they score well on the caption evaluation.

[4] Potapov *et al.* (2014)

[5] Zhang *et al.* (2016)

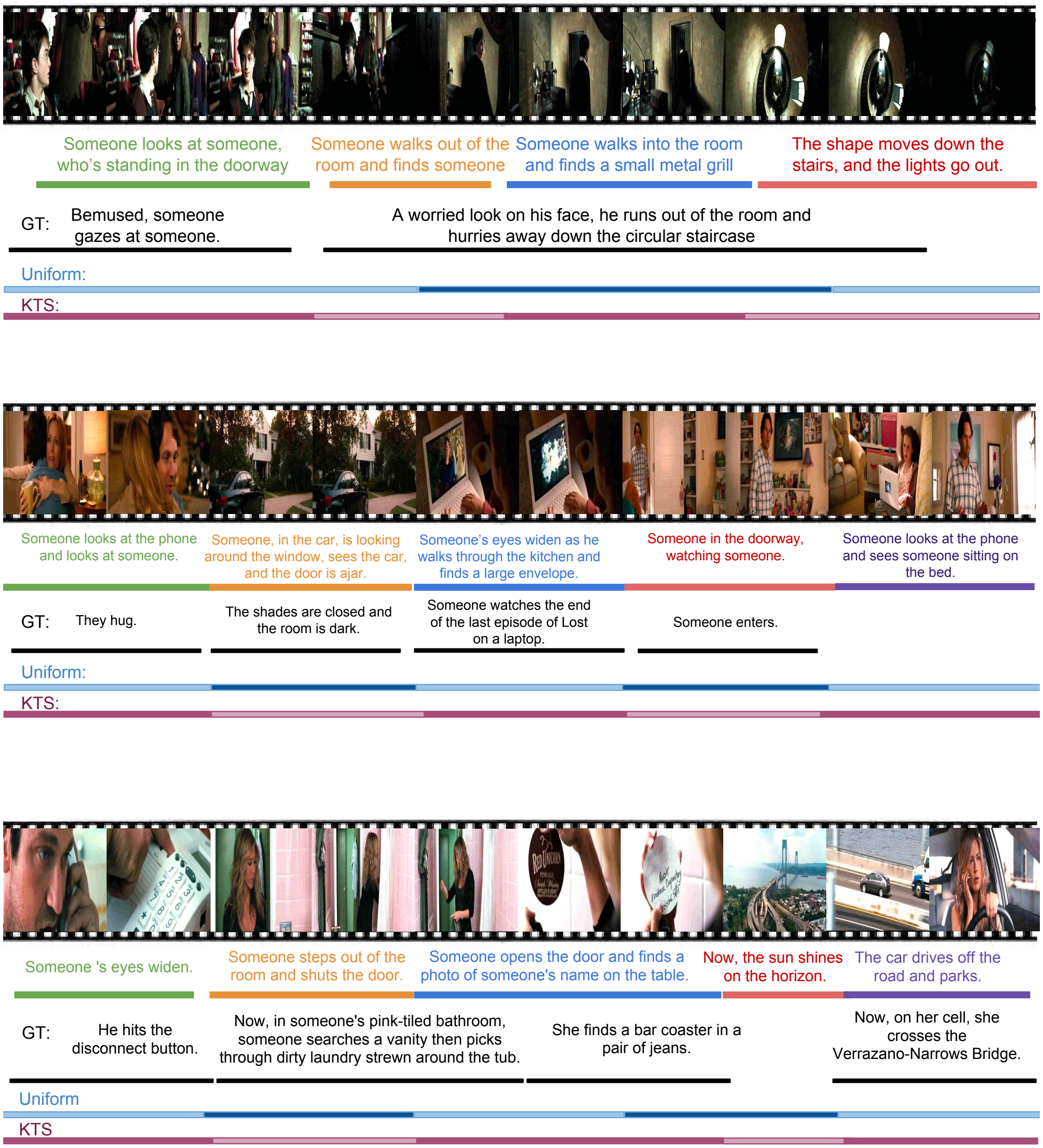

Someone looks at someone, who's standing in the doorway
Someone walks out of the room and finds someone
Someone walks into the room and finds a small metal grill
The shape moves down the stairs, and the lights go out.
GT:
Bemused, someone gazes at someone.
A worried look on his face, he runs out of the room and hurries away down the circular staircase
Uniform:
KTS:
Someone looks at the phone and looks at someone.
Someone, in the car, is looking around the window, sees the car, and the door is ajar.
Someone's eyes widen as he walks through the kitchen and finds a large envelope.
Someone in the doorway, watching someone.
Someone looks at the phone and sees someone sitting on the bed.
GT:
They hug.
The shades are closed and the room is dark.
Someone watches the end of the last episode of Lost on a laptop.
Someone enters.
Uniform:
KTS:
Someone 's eyes widen.
Someone steps out of the room and shuts the door.
Someone opens the door and finds a photo of someone's name on the table.
Now, the sun shines on the horizon.
The car drives off the road and parks.
GT:
He hits the disconnect button.
Now, in someone's pink-tiled bathroom, someone searches a vanity then picks through dirty laundry strewn around the tub.
She finds a bar coaster in a pair of jeans.
Now, on her cell, she crosses the Verrazano-Narrows Bridge.
Uniform
KTS

Figure 7.3: Example (foreground) segments and descriptions generated by our model on the movies from MPII-MD and MVAD. The top rows show input video frames. Immediately below the video frames is the output description and segmentation of our model, followed by the groundtruth (GT) description and segmentation (in black). The bottom two rows show the output of the uniform and KTS segmentation approaches (the lighter and darker shades represent alternate segments).

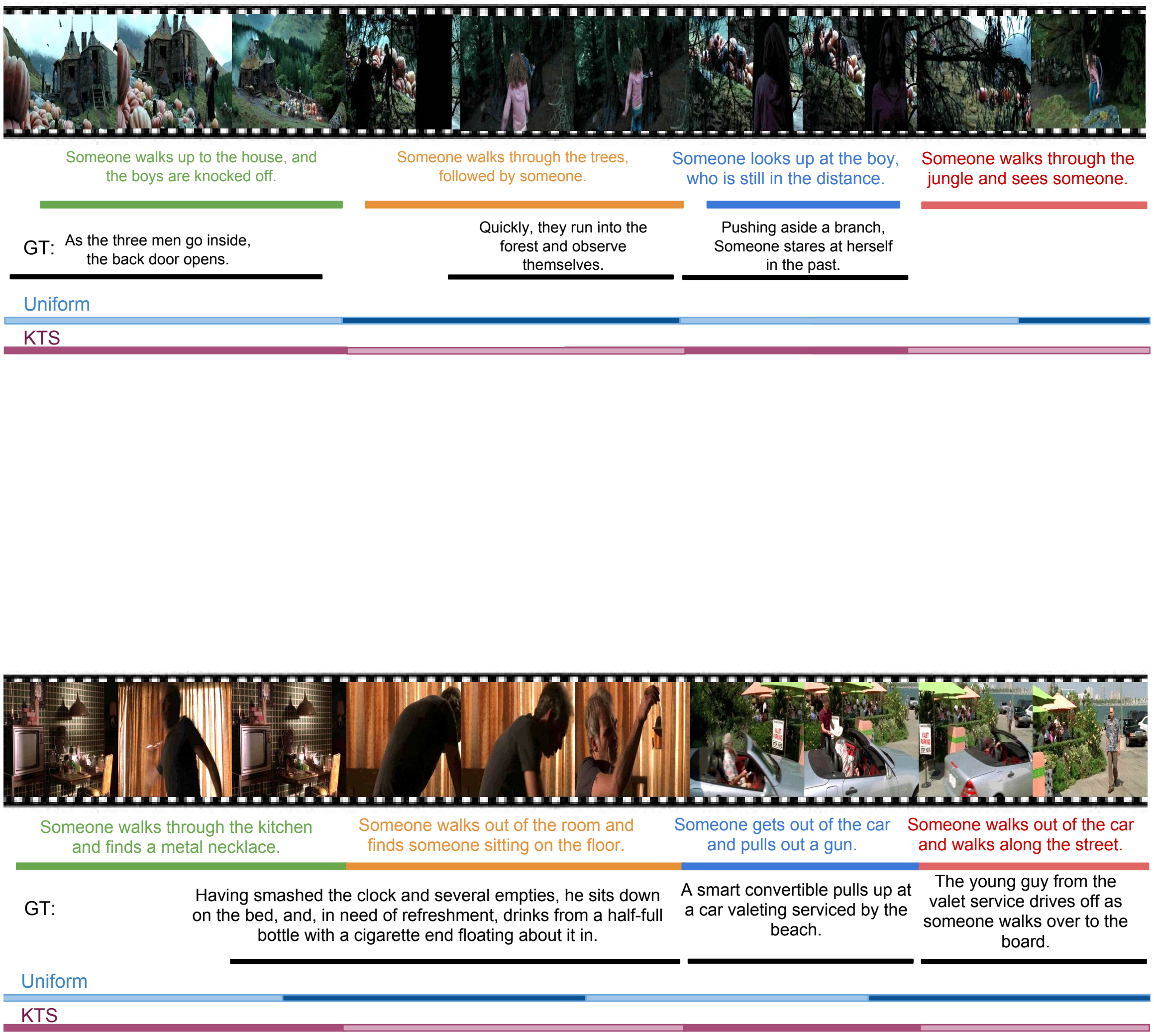

Someone walks up to the house, and the boys are knocked off.
Someone walks through the trees, followed by someone.
Someone looks up at the boy, who is still in the distance.
Someone walks through the jungle and sees someone.
GT:
As the three men go inside, the back door opens.
Quickly, they run into the forest and observe themselves.
Pushing aside a branch, Someone stares at herself in the past.
Uniform
KTS
Someone walks through the kitchen and finds a metal necklace.
Someone walks out of the room and finds someone sitting on the floor.
Someone gets out of the car and pulls out a gun.
Someone walks out of the car and walks along the street.
GT:
Having smashed the clock and several empties, he sits down on the bed, and, in need of refreshment, drinks from a half-full bottle with a cigarette end floating about it in.
A smart convertible pulls up at a car valeting serviced by the beach.
The young guy from the valet service drives off as someone walks over to the board.
Uniform
KTS

Figure 7.4: Example (foreground) segments and descriptions generated by our model on the movies from MPII-MD and MVAD. The top rows show input video frames. Immediately below the video frames is the output description and segmentation of our model, followed by the groundtruth (GT) description and segmentation (in black). The bottom two rows show the output of the uniform and KTS segmentation approaches (the lighter and darker shades represent alternate segments).

7.5.1 Discussion

We note here that our method can be extended directly to describe full movie-length videos. Although our TSDN model can only processes 1 to 3 minute clips at a time, the DVS segments themselves are significantly smaller, just a few seconds. Hence, we can first apply the unsupervised KTS change-point detection algorithm to clip the movies at change-points that are one or two minutes apart producing reasonably sized 1-2 minute long clips. We can then apply our TSDN approach to identify and describe foreground segments within each of the clips, to generate descriptions for the entire movie. However, evaluating this approach on full movies can be tricky since it introduces additional sources of error from generating 1-2 minute clips.

7.6 Conclusion

In this chapter, we formulated a new and challenging video understanding task, activity description in unsegmented movies. We proposed an approach for solving this problem based on convolutional and recurrent layers that learns to segment the video temporally selecting salient activities, and describe them using an encoder-decoder architecture. As this is a novel task, there are no existing methods that can be directly applied for comparison, so we compare our model to several straightforward baselines that use bottom-up segmentation of video and caption produced segments. Based on extensive evaluations on two large-scale DVS movie datasets, we demonstrate that our model is able to select and describe segments in long videos.

Chapter 8

Related Works

In this chapter we look at related works that can help place this dissertation in context of the overall progress within the sub-field of image and video description. I will overview relevant research works, many of which were either contemporaneous to the approaches presented in this thesis, or subsequently built upon and improved the architectures and models seen in earlier chapters. We will first briefly overview approaches to deep image captioning that have formed the basis of some of the works in this thesis (Chapters 3 and 6). We then look at developments in video description, in particular different categories of improvements that build upon models from Chapters 3 and 4 of this thesis. We will also look at works that incorporate external knowledge to improve image and video description for specific classes of problems, in particular for describing unseen and novel object categories and scenes in images and movies. Finally, we look at techniques that have been developed to handle some of the challenges in understanding longer videos, including video summarization, activity localization, and description in longer videos.

8.1 Deep Image Captioning

As mentioned in Chapter 2, two main factors have contributed to the growing research in image captioning: 1) advances in deep neural networks, and 2) development of large corpora of images with text descriptions (Hodosh *et al.*, 2014, Ordonez *et al.*, 2011, Lin *et al.*, 2014). Deep neural network models have gained tremendous popularity for both their performance and potential for end-to-end training in both computer vision and NLP. Deep image captioning approaches were the first to combine the power of both convolutional neural networks as well as recurrent neural networks. These captioning models work by first encoding an image into a fixed length feature vector using a CNN, and then generate a description by either conditioning text generation on image features (Donahue *et al.*, 2015, Karpathy and Fei-Fei, 2015, Vinyals *et al.*, 2015) or by embedding image features and previously generated words into a multimodal space (Kiros *et al.*, 2015, Mao *et al.*, 2014) before predicting the next word. For caption generation specifically, RNNs have been a popular choice. While Karpathy and Fei-Fei (2015) and (Mao *et al.*, 2014) used a simple RNN to generate the description, Donahue *et al.* (2015), Kiros *et al.* (2015) and Vinyals *et al.* (2015) used LSTMs which outperformed simple RNNs on the task. However there have also been some works that have favored log bilinear language models (Kiros *et al.*, 2015) and maximum entropy language models (Fang *et al.*, 2015) as well.

For the visual representation itself, most models represent images with an intermediate representation from a convolutional neural network (such as the activations of the fully-connected layer just before classification), since these features, although trained for object recognition, generalize well to other tasks. However, there are a few models that represent images as a vector of confidences over a fixed number of visual concepts (Fang *et al.*, 2015, Hendricks *et al.*, 2016) and this representation can be particularly advantageous if the domain is well specified. In almost all cases, the parameters of the visual pipeline are initialized with weights trained on the ImageNet Large Scale Visual Recognition Challenge Russakovsky *et al.* (2015), which we also did in all of our models (detailed in Chapter 3, Section 3.3). A very interesting variation of the visual representation in image captioning has been the introduction of "attention" (Xu *et al.*, 2015b) where the model doesn't just have a single fixed representation of the image, but instead learns a spatially weighted representation of the image which focuses on different locations in the image as it is generates each word of the description.

8.1.1 Multiple descriptions

All of the works mentioned previously focus only on generating a single description for each image. There have also been works that have looked at generating multiple descriptions for images. These can be categorized broadly into two related lines of research 1) generating multiple descriptions i.e, more detailed descriptions, for a single image, and 2) generating descriptions for a sequence of related images.

Multiple descriptions for an image. Johnson *et al.* (2016) introduce the task of dense captioning which aims at describing different regions in the image with a phrase or a sentence. They introduce a CNN-LSTM model that detects and describes different regions of interest within a single image. While "attention" based image-captioning models (Xu *et al.*, 2015b) are also capable of examining different regions in the image, they only generate a single description. Whereas, Johnson *et al.* (2016) explicitly propose and select regions of interest in the image using object detection approaches (Girshick *et al.*, 2014, Ren *et al.*, 2015b) and then use an LSTM decoder to generate descriptions for each region. Our TSDN model in Chapter 7 is a video analog of the same task since our goal is to identify spatio-temporal regions of interest in the video and generate descriptions.

Although, the model in Johnson *et al.* (2016) generates multiple descriptions for an image, these descriptions are independent and do not form a coherent whole. Krause *et al.* (2016) addresses the task of generating a coherent paragraph about a single image by using a hierarchical recurrent neural network decoder when generating the description. Specifically, they also first identify regions of interest in the image (Girshick *et al.*, 2014) and they then project and pool these to generate a compact encoded representation. The first level in their hierarchical RNN decoder determines how many sentences to generate along with the topic words for each (conditioned on the visual representation), and a subsequent RNN (one for each sentence) consumes these topic words to generate sentences.

Describing a sequence of images. This line of research to generate multiple descriptions for images extends directly to the idea of generating or selecting a set of coherent sentences to describe a sequence of related images. Park and Kim (2015) addresses this task by introducing an explicit coherence model that combines the visual representation of images from a CNN encoder and text representation from a bi-directional RNN to produce sentences that form a coherent description of the image sequence.

8.2 Deep Video Description

While the goal in describing a sequence of images was to generate multiple descriptions, with short videos the inherent temporal coherence in frames represent a single event that needs to be described. Since video description is a natural extension of image captioning the developments in deep video description have been directly influenced by growth in deep image captioning. The model in Chapter 3 (Venugopalan *et al.*, 2015b) was the first fully deep video captioning model and it was inspired by the image captioning models in Donahue *et al.* (2015) and Vinyals *et al.* (2015). However, this model simply temporally pooled features from all frames, ignoring the temporal sequence entirely. While the S2VT model in Chapter 4 proposed ways of modeling the temporal sequence using LSTM encoders and optical flow (Brox *et al.*, 2004), contemporaneous work by Yao *et al.* (2015) developed models to focus on relevant temporal segments to generate descriptions. The models by Yao *et al.* (2015) and subsequent models focused on generating a better visual representation of the videos. The techniques proposed by these works fall into two broad classes contributions, 1) "*attention*" and 2) hierarchical models. We will review these and few other improvements next.

8.2.1 Attention-based Models

As mentioned in the previous section, the concept of "attending" to different regions in the image was proposed by Xu *et al.* (2015b). This concept can be extended to videos in two ways, either as just "temporal attention" attending to select frame sequences (Yao *et al.*, 2015, Pan *et al.*, 2016), or as "spatio-temporal attention" focusing on different spatio-temporal volumes of the video (Yu *et al.*, 2016). Yao *et al.* (2015) proposed a soft-attention mechanism that learns to weight the frame features in order to create a different representation of the video as it decodes and generates each word in the description. This in turn allows the model to focus on different portions of the frame sequence as it describes the video. Additionally, they also introduce a 3-D CNN to learn features that can incorporate both spatial and temporal association in video frames. Yu *et al.* (2016) also learn to process spatio-temporal volumes in the video by learning object detectors to focus on individual spatial regions in the image. They track these regions across frames attending to different spatio-temporal sections of the video to improve the visual representation.

8.2.2 Hierarchical Models

Another way to generate better visual representations is by creating a hierarchical architecture to encode the video frames (Pan *et al.*, 2016, Yu *et al.*, 2016) as opposed to processing all items in the input sequence in a similar manner (as seen in the S2VT model in Chapter 4). The model in Pan *et al.* (2016) consists of 2 LSTM layers like S2VT, however, while their first layer processes all frames, their second layer gets inputs from the first layer at uniform intervals creating a temporal depth in the network that process the video in chunks to generate a better representation. Additionally, Pan *et al.* (2016) also incorporate temporal-attention in their models to further improve performance. Yu *et al.* (2016), however, use a hierarchical RNN when decoding and generating the description.

8.2.3 Different RNN and CNN features

Ballas *et al.* (2016) explore the use of activation maps from different layers in the CNN to generate better visual representations for video object and activity description. Similarly, Xu *et al.* (2015a) use multi-scale CNN features to recognize objects of different scales (magnifications) in videos. The models in Yao *et al.* (2015) and Ballas *et al.* (2016) also study the use of Gated Recurrent Unit (GRU) RNNs as opposed to LSTM RNNs. GRUs are a simpler more memory efficient variant of LSTMs with just a single update or reset gate that helps address the problem of vanishing gradients to learn long term dependencies. It's performance has shown to be similar to LSTMs in language modeling tasks Chung *et al.* (2014). Additionally, the models by Yao *et al.* (2015) and Ballas *et al.* (2016) which use GRUs have performed competitively with our own LSTM captioning models in Chapter 4.

8.2.4 Multi-modal Video Captioning

Ramanishka *et al.* (2016) extends our S2VT approach to incorporate audio information along with the visual and text modality when generating descriptions. Specifically, they use the popular audio feature - Mel Frequency Cepstral Coefficients (MFCC) which have been used widely in various audio processing tasks such as automatic speech recognition, music transcription, and environment classification (Hinton *et al.*, 2012, Giannakopoulos, 2015, Beritelli and Grasso, 2008).

Their model concatenates the audio features along with the visual information and provides it as input when encoding the video. Their captioning decoder model is similar to S2VT.

8.3 Use of External Resources for Captioning

Another way to improve visual and textual representations learned by visual description models is by incorporating external knowledge as seen in Chapters 5 and 6. There have been several works in image and video description that have used external knowledge to address interesting problems such as, recognizing or describing rare/unseen objects (Mao *et al.*, 2015, Hendricks *et al.*, 2016), identifying characters in movies and TV shows from scripts and dialogs (Everingham *et al.*, 2006, Haurilet *et al.*, 2016), aligning and generating descriptions for movies from book chapters (Tapaswi *et al.*, 2015, Zhu *et al.*, 2015) or cooking videos Malmaud *et al.* (2015) or just for improving descriptive quality (Yang *et al.*, 2011, Thomason *et al.*, 2014). This section will overview a few approaches that are particularly relevant to our work in Chapters 5 and 6.

8.3.1 Captioning Novel Objects

In Chapter 6 we use external knowledge to describe novel objects in images. Mao *et al.* (2015) was one of the first deep captioning models to look at this problem. They proposed an approach that extends a model's capability to describe a small set of novel concepts (e.g. *quidditch, samisen*) from a few paired training examples while retaining its ability to describe previously learned concepts. On the other hand, Hendricks *et al.* (2016) introduce a model that can describe many objects already existing in English corpora and object recognition datasets (ImageNet) but not in the caption corpora (e.g. *pheasant, otter*). Our work in Chapter 6 focused and built on the latter case. Hendricks *et al.* (2016) integrate information from external text and visual sources, and explicitly transfer ('copy') parameters from objects seen in image-caption data to unseen ImageNet objects to caption these novel categories. While this works well for many ImageNet classes it still limits coverage across diverse categories and cannot be trained end-to-end. Furthermore, their model cannot caption objects for which few paired training examples already exist. Our NOC framework (Chapter 6) integrates distributional semantic embeddings implicitly, obviating the need for any explicit transfer and making it end-to-end trainable. It also extends directly to caption ImageNet objects with few or no descriptions.

8.3.2 Describing Scenes in Movies

Our work in Chapter 3 used external image captioning data to aid video captioning and our model in Chapter 5 used monolingual text data to improve grammaticality. But even prior to these, Yang *et al.* (2011), Krishnamoorthy *et al.* (2013), Thomason *et al.* (2014) all used external text sources to improve descriptive quality for videos. Movie description is another specific sub-domain where the use of external resources have played an important role in enhancing descriptions as a whole. Rohrbach *et al.* (2015a) used scene datasets (Xiao *et al.*, 2010) and action labels from Wang *et al.* (2011) to improve descriptions for movies. Tapaswi *et al.* (2015), Zhu *et al.* (2015) both used information from books to align scenes in videos to book chapters, there by captioning movie scenes with paragraphs from the book. There have also been works that have used external resources such as movie scripts to identify character names in TV shows and movies (Everingham *et al.*, 2006).

8.4 Long Videos

Our work in Chapter 7 that segments and describes longer videos, builds on several sub-areas of research within video understanding. In particular, developments in temporal segmentation (Potapov *et al.*, 2014, Poleg *et al.*, 2014) play an important role in identifying coherent sequence of frames in long videos and segmenting them. Additionally, work in video summarization and activity localization Baraldi *et al.* (2016), Zhang *et al.* (2016), Escorcia *et al.* (2016) help building architectures that can process and recognize salient activities and events in long videos.

8.4.1 Video Segmentation and Summarization

In video summarization, the goal is to generate a shorter video that highlights important events (Lu and Grauman, 2013) or important objects (Lee and Grauman, 2015). This involves selecting frames/shots using unsupervised or supervised methods. Most works on video summarization first pre-process videos by generating temporal segments. In temporal segmentation, the goal is to break up a long video into meaningful chapters, such as for browsing egocentric video Poleg *et al.* (2014). To temporally segment long cooking videos, Rohrbach *et al.* (2014) uses agglomerative clustering based on attribute classifier similarity which consists of object and activity classifiers. An additional background classifier is used to reject segments which are irrelevant or noisy. While some segmentation methods rely on shot boundaries, Potapov *et al.* (2014) proposes to also detect general change points, including changes within shots, by comparing all pairs of frames. Once the video is thus segmented, they also classify the segments by their importance to generate a summary. While Potapov *et al.* (2014) use a kernel based method to detect changes, Baraldi *et al.* (2016) process long TV length shows by training a siamese architecture to identify sequences of frames that are temporally coherent and detect changes. They then generate a summary by selecting some frames from each segment. Zhang *et al.* (2016) performs video summarization by training a bi-directional LSTM to predict which frames should be included in the summary. Our work in Chapter 7 combines approaches from Potapov *et al.* (2014) and Zhang *et al.* (2016) and uses LSTMs to predict foreground/background segments, but our goal is description rather than visual summarization.

8.4.2 Object and Activity Localization

The task of identifying salient events worth describing is also closely related to action and object detection in video. While the goal in detection is to identify salient regions and generate a single label, we are interested in generating a complete sentence. Research in object detection aims to find object boundaries while identifying their categories. Recent methods like Faster RCNN (Ren *et al.*, 2015b) propose object-like regions based on CNN features, and then classify these foreground proposals to predict labels. The method in Johnson *et al.* (2016), as described before (Section 8.1.1) goes beyond category detection to also describe focused regions in the image. Our work in Chapter 7 is analogous to this in the sense that our goal is to localize and describe events but in long videos.

Compared to object detection, activity detection in video is less explored due to the scarcity of annotated data as well as the complexity of the problem. Recent methods focus on fixed action categories and identify the start and end time of each action using multi-scale sliding window search (Shou *et al.*, 2016, Singh *et al.*, 2016), or propose temporal segment end points (Escorcia *et al.*, 2016, Yeung *et al.*, 2016, Ma *et al.*, 2016). Similar to our work, the latter methods also identify temporal segment proposals likely to contain actions, but predict only fixed action categories like "playing hockey" or "wrapping presents." In contrast, Chapter 7 looks at generating fluent descriptive sentences. Moreover our definition of an event/activity comes from the domain as DVS as something that is worth describing for the visually impaired particularly when taking context into account.

8.4.3 Captioning Long Videos

While most existing work on video description has been on generating single sentence description, work by Yu *et al.* (2016) introduced a paragraph-RNN (p-RNN) model to generate descriptions for longer videos, specifically multi-sentence paragraph captions. They use a hierarchical RNN model to process visual input and generate multi-sentence descriptions. However, they do not localize events in the video, whereas we aim to temporally segment *and* describe longer videos. Similar to our work in Chapter 7, Shin *et al.* (2016) and Baraldi *et al.* (2017) propose methods to segment videos and movies and generate descriptions, however, they use the original short clips from the MPII-MD and M-VAD datasets which only contain 4-6 second segments, not longer continuous clips with multiple sentences as we do.

Recent contemporaneous work by Krishna *et al.* (2017) is the most relevant to our work in Chapter 7. Krishna *et al.* (2017) also look at segmenting salient events in longer videos and describing them. They introduce a new dataset of annotations for ActivityNet dataset (Caba Heilbron *et al.*, 2015), which is a collection of Youtube video clips for activity recognition and localization. Similar to our work, Krishna *et al.* (2017) also use existing methods to generate segments. While we use unsupervised methods (Potapov *et al.*, 2014), they use supervised activity localization methods (Escorcia *et al.*, 2016). Also, unlike our Bi-LSTM captioning frame work, they use a hierarchical RNN with attention to generate captions. Our view of the task also differs slightly in the sense that while Krishna *et al.* (2017) aim to describe generic event sequences in videos our work focuses on movie DVS.

8.5 Summary

In summary, there have been several recent works in image and video captioning that are closely related to the task and methods presented in this dissertation. While some of the research in image captioning have inspired the video and image description works presented in this thesis, many of the video description works have built on models introduced here. In particular we discussed related work that presented improved visual representation using “attention”, hierarchical approaches, as well as better CNN representation and RNN variants. We also looked at other methods relevant to our own that used external resources to enhance image and video descriptions, particularly movie descriptions. Finally, we saw several methods for localizing events in longer videos, identifying salient sub-shots/frames to summarize, as well as methods to generate multiple descriptions for longer videos.

Chapter 9

Future Directions

In this chapter we briefly outline some of the future directions in which the research in this thesis can be extended. While work presented in this dissertation looked at addressing some of the challenges in video description, specifically describing open-domain videos, with large vocabularies, variety of objects, scenes and activities, as well as generating descriptions for events in movies, it is still limited in ways. Most of the work in this thesis looks at generating descriptions of distinct and independent events in videos, one immediate direction of extension is to be able to generate descriptions in context of what has already been described and also take pragmatics into account when generating descriptions. Another direction that has potential applications in video retrieval and understanding is generating textual summaries of long videos. Yet another promising direction is to enhance movie descriptions. As seen in Chapters 5 and 7, work presented in this thesis is only a small initial step towards automatically generating DVS descriptions, and our contributions can be improved in many ways. Here we briefly look at some of the challenges and sketch steps that one can take to extend work in each of these directions.

9.1 Describing Video Events in Context

Almost all work in video description has looked primarily at generating descriptions for independent events. With short video captioning, techniques were built for generating a simple description of the event, and even in case of longer movie clips, the focus was more on describing a salient event. While this view of captioning is suitable for some applications like generating descriptions for the visually impaired, identifying and articulating multiple events in videos can also enable better methods for video retrieval and potentially video question answering. However the challenge here is to identify multiple events, and generate coherent descriptions in context. Work presented in Chapter 7 can be extended in two primary directions to enable this.

9.1.1 Joint Localization and Description

While our TSDN method proposes segments and generates descriptions, it has two main drawbacks 1) it does not recognize multiple (potentially concurrent/overlapping) events, 2) it does not address these tasks jointly, i.e, the segmentation and captioning module do not interact to learn a jointly useful representation. One way to address these in TSDN is to potentially have a multi-task network where the visual representation is trained based on the loss from both temporal segmentation as well as description. Specifically, we can introduce a deep model that proposes temporal segments (similar to Escorcia *et al.* (2016)) and share weights of the visual encoding module i.e., the Bi-LSTM encoder, with both the segment proposal module and the LSTM model for captioning. Such a model would have two loss functions or objectives, one coming from the segment proposal network and another coming from the caption RNN. The training strategy for such a model would be quite similar to our Novel Object Captioning approach in Chapter 6. In this method, although the segment proposal and captioner network don't directly interact, the loss from both modules will force the model to learn a better visual representation that informs both tasks.

Another way of addressing this task could be analogous to Johnson *et al.* (2016). Johnson *et al.* (2016) propose a single end-to-end model that is capable of both jointly identifying regions in the image and generating descriptions for each. Here the visual representation is used to first generate bounding box proposals and a confidence score for each proposal. Descriptions are generated for proposals with high confidence scores. In this case, the key difference between this model and a multi-task model is that the loss from the captioning model is propagated back to the visual encoder through the segmentation module. So, the parameters of the segmentation network are informed by the captioning network.

9.1.2 Coherent Descriptions for Movies

Paragraph descriptions for sequences in video. Yet another direction one can take for describing events in longer videos is to focus on the language model to generate coherent descriptions of multiple events. The previous chapter discussed works that generated coherent descriptions for a single image (Krause *et al.*, 2016), for a sequence of images (Park and Kim, 2015) as well as for domain specific events in video (Yu *et al.*, 2016). Both Krause *et al.* (2016) and Yu *et al.* (2016) propose hierarchical models to generate a coherent paragraph based on the visual content. While Krause *et al.* (2016) propose to first identify how many sentences to describe and the topics of each, Yu *et al.* (2016) focus on describing different temporal sequences/events in the video, thus generating descriptions of different events that form a coherent paragraph. The key novelty of these approaches lies in the hierarchical captioning module. Similar hierarchical RNN decoding approaches can be combined with the temporal segmentation network to generate coherent descriptions in movies.

Inference-driven Pragmatics for context-specific descriptions. Captions generated by S2VT and the Bi-LSTM models for consecutive clips in the movie datasets are quite similar and sometimes identical. One can consider generating context-specific descriptions by combining learned semantics with an inference-driven approach to pragmatics (Andreas and Klein, 2016, Vedantam *et al.*, 2017). The work of Andreas and Klein (2016) and Vedantam *et al.* (2017) look at generating discriminative captions for a given target image presented with context in the form of another image having similar content. The goal in their task is to generate a caption that can help an observer distinguish (and select) the target image when presented with both the target and context images. While Andreas and Klein (2016) sample multiple sentences from the image-caption decoder to produce a caption that can be discriminative, Vedantam *et al.* (2017) present a simple approach that modifies the distribution of the decoder (specifically Equation (3.2)) based on the language model's distribution conditioned on the target image as well as the context image representations. Both of these approaches can be applied to the video description models presented in this work. In particular, for generating descriptions on the movie corpora. By using an inference-driven approach and considering the previously described event/segment as context, the model can focus on generating a description that can provide new information about the current target event sequence.

9.2 Generating Textual Summaries of Long Videos

Works in video summarization (Lu and Grauman, 2013, Lee and Grauman, 2015, Zhang *et al.*, 2016, Potapov *et al.*, 2014) have focused predominantly on generating visual summaries of clips given specific constraints on the length of the summary. These clips could be short 3-5 minute clips from typical events like birthday parties that need to be reduced to 30 seconds or could be several hour long ego-centric videos or cooking videos that need to be compressed to contain just key moments. An interesting variation from the perspective of description would be to generate textual summaries of long videos.

Recent work by Sah *et al.* (2017) presents a simple approach building upon the work in S2VT for this task. Specifically, they first generate a visual summary of 6-8 hour-long ego-centric videos by ranking and selecting coherent subshots. They then apply the S2VT captioning approach to generate descriptions for each subshot in the visual summary. Finally, they use off-the-shelf text summarization techniques (Landauer, 2006, Erkan and Radev, 2004) to generate a textual summary of the video.

While the approach in Sah *et al.* (2017) is simple, the S2VT or the TSDN models can be modified and extended in several ways to generate both the visual and textual summary of the video. One variation would be to build a temporal segmentation and summarization network, to generate visual segment representations, as well as text topics that can be combined with a hierarchical RNN decoder such as the one from Yu *et al.* (2016) or Krause *et al.* (2016) to generate a paragraph summary of the long video clip. Constraints can be placed both on the visual modules (restricting video duration) and the text modules (restricting word or character length) to generate summaries of different lengths. While processing long videos can be quite challenging recent advances in video summarization (Zhang *et al.*, 2016, Gygli *et al.*, 2016) and text summarization (Durrett *et al.*, 2016, Nallapati *et al.*, 2016) can be combined to generate textual summaries of long videos.

9.3 Enhancing Movie Description

We can 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 (Venugopalan *et al.*, 2015a, Yao *et al.*, 2015) 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 (Everingham *et al.*, 2006, Apostoloff and Zisserman, 2007). 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 Cour *et al.* (2009; 2011) to learn characters from ambiguously or partially labeled images.

9.3.1 Identifying Character Screen Presence from Movie Scripts

There is a body of prior work on identifying characters in video streams, e.g.,Everingham *et al.* (2006), Apostoloff and Zisserman (2007), 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 (Malmaud *et al.*, 2015, Zhu *et al.*, 2015, Tapaswi *et al.*, 2015). However, in the description task 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. One can 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 (Cour *et al.*, 2009; 2011). 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.

9.3.2 Associating Character Names with Actors

Another technique to get names of characters is by using face recognition to identify the different actors, and use screen credits to associate actors to their characters on screen. Even without an actor recognition model, we can employ semi-supervised approaches to identify characters/actors (Cour *et al.*, 2011) in movies by simply 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 (Tomasi and Kanade, 1991). 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. A simpler option would also be to use our Novel Object Captioning approach from Chapter 6, to incorporate character names into the sentences.

Alternately, we can also employ simple language substitution techniques to completely avoid annotating some sentences with character names on the test movies and instead use placeholders. 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 replace specific placeholders using modification of neural checklist models (Kiddon *et al.*, 2016) or other RNNs that can copy/replace words (Gu *et al.*, 2016), these can be used at test time to substitute a place holder for the actual character name.

9.3.3 Fully-Automating DVS

Recall that DVS is a separate *audio track* for the visually impaired. Hence, fully-automating DVS descriptions requires addressing two key challenges: 1) it is necessary to incorporate multi-modal context specifically, both video as well as audio (speech and sounds) to capture the entire context during description, additionally, 2) we need to be able to insert the generated descriptions interleaving them with the existing audio track at suitable locations.

Multi-modal context for DVS. In the case of DVS, current video description models rely exclusively on the visual input for generating captions. Video representations that include audio signals along with the video frames can provide more useful contextual information and also potentially enhance description quality. A natural extension to the video models in this work would be to consider multi-modal context from both the audio and the video information when generating DVS descriptions for movies. This can be accomplished by either 1) applying speech recognition techniques (Hinton *et al.*, 2012, Graves and Jaitly, 2014) and converting the speech to a text modality or 2) by using the audio signals directly, as in Ramanishka *et al.* (2016) to generate audio feature inputs. The advantage of the latter is that we would allow the model to learn features from raw audio which can provide more information e.g., background music could be indicative of emotions/mood. The resulting audio representation from either of these approaches can then be combined to produce a multi-modal representation of the video for generating DVS.

Interleaving descriptions with audio. While identifying events that are worth describing to the visually impaired is a major challenge in itself, once we have identified and generated a description, positioning it appropriately is also a very hard problem. To fully-automate DVS it is necessary to insert the generated description at appropriate locations within the existing audio track of the movie. This introduces many new constraints such as, (i) the length or duration of the description, (ii) the original location where the event occurs, (iii) identifying an appropriate position in the existing audio track that is free of dialogues in order to insert the DVS. It might also be necessary to rephrase or potentially shorten descriptions to fit within a location that is sufficiently close to where the event actually takes place. Some techniques from video summarization could potentially help in addressing some of these challenges. To clarify, one could look at modifying summarization techniques (Lu and Grauman, 2013, Lee and Grauman, 2015, Potapov *et al.*, 2014) to explicitly identify segments in the video where we can insert the DVS instead of identifying salient events that need to be included in the summary. Here again, we need to express the constraints of inserting the DVS into an appropriate formalism to address the task. This could be an interesting and challenging direction of exploration that could complete this work.

Chapter 10

Conclusion

Generating natural language descriptions for events in videos enables several applications. The last few years have seen a dramatic interest in description of static images and growing interest in video description. This thesis focused on developing methods for generating natural language descriptions that capture sequences of activities depicted in diverse video corpora, where limited prior work exists.

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. With these aims in mind this thesis looked to address some of the major obstacles in video description, namely, 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 dissertation presented the first fully deep models (Chapters 3 and 4) 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. We demonstrated the versatility of our approaches by evaluating them on clips from open-domain Youtube videos as well as clips from commercial movies.

This work also presents several methods to significantly extend research in this area. Specifically, Chapter 5 presented strategies to generate more diverse and accurate descriptions by integrating prior linguistic knowledge. In addition, Chapter 6 introduced an end-to-end deep model to describe novel objects unseen in paired image-caption data. While previous deep image captioning techniques were restricted to describing a small set of object categories within existing image-caption corpora, our proposed method is able to describe hundreds of object categories that can be identified by modern deep object recognition approaches.

In addition to these, to make video description useful in real world applications such as generating descriptions for the visually impaired, it is necessary to be able to process and describe a continuous stream of videos. Chapter 7 takes a step in this direction and introduced the task of temporally segmenting and describing longer videos. Specifically, we presented techniques to process longer multi-activity videos by learning to segment and describe coherent event sequences in full-length movies. Our task was aimed at automatically generating DVS descriptions for the visually impaired. In addition to proposing new methods, we presented appropriate datasets for the task and evaluated our methods on these.

Chapter 8 overviews many relevant and related works placing the contributions of this thesis in context. Finally, Chapter 9 looked at some of the remaining challenges and sketched steps to extend the work presented here.

Language and Vision is a rapidly growing area of research. This thesis presented techniques to address some of the challenges in this emerging field. We hope that some of the ideas and insights presented here will be useful in future works.

Bibliography

Jacob Andreas and Dan Klein. Reasoning about pragmatics with neural listeners and speakers. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 106

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

N. E. Apostoloff and A. Zisserman. Who are you? – real-time person identification. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2007. 108

H. Aradhye, G. Toderici, and J. Yagnik. Video2text: Learning to annotate video content. In *International Conference on Data Mining Workshops (ICDMW)*, 2009. 6

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3, 8, 15

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *International Conference on Learning Representations (ICLR)*, 2016. 98

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Recognizing and presenting the storytelling video structure with deep multimodal networks. *IEEE Transactions on Multimedia*, 2016. 77, 100, 101

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 102

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 *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012. 1, 6

Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research (JMLR)*, 3:1107–1135, 2003. 5

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 848–854, 2004. 5

F Beritelli and R Grasso. A pattern recognition system for environmental sound classification based on mfccs and neural networks. In *International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–4, 2008. 98

Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 25–36, 2004. 26, 29, 97

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. 102

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. 7

David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011. 16, 19, 27, 30, 39, 44

Xinlei Chen and Lawrence C Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2422–2431, 2015. 3

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. 32

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014. 8, 10, 14

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. Presented at the Deep Learning workshop at NIPS2014. 98

Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 108, 109

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research (JMLR)*, 12:1501–1536, 2011. 108, 109

P. Das, R. K. Srihari, and J. J. Corso. Translating related words to videos and back through latent topics. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM)*, 2013. 1, 6

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 6

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. 7

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. 5

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 2, 56

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3450–3457, 2012. 7

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2014. 20, 31, 44, 57, 85

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 *ACM International Conference on Multimedia Retrieval (ICMR)*, 2012. 1, 6

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. In *IEEE International Conference on Machine Learning (ICML)*, volume 32, pages 647–655, 2014. 15

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3, 5, 8, 10, 11, 13, 14, 15, 26, 29, 40, 48, 58, 75, 77, 94, 95, 97

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(4):677–691, 2017. 16

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. 107

Desmond Elliott and Frank Keller. Comparing automatic evaluation measures for image description. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014. 21

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004. 107

Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 768–784. Springer, 2016. 75, 100, 102, 103, 105

M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2006. 99, 100, 108

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1473–1482, 2015. 3, 5, 48, 95

A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 1, 5

D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2011. 5

Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS one*, 10(12):e0144610, 2015. 98

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 81, 96

G. Gkioxari and J. Malik. Finding action tubes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 29

A Gorban, H Idrees, Y-G Jiang, A Roshan Zamir, I Laptev, M Shah, and R Sukthankar. THUMOS Challenge: Action Recognition with a Large Number of Classes. http://www.thumos.info/, 2015. 74

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *IEEE International Conference on Machine Learning (ICML)*, 2014. 8, 14, 110

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649, 2013. 3

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. 110

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shoot recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2, 6, 7, 15, 19, 20

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. 5

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. 39, 41

Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1001–1009, 2016. 107

L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12(10):993–1001, Oct 1990. 43

Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen. Naming TV Characters by Watching and Analyzing Dialogs. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016. 99

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 48, 74

Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, and Trevor Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 49, 50, 55, 56, 57, 58, 59, 61, 62, 64, 66, 69, 95, 99

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. 98, 110

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8), 1997. 3, 10, 13

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 10

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 *Transactions of the Association for Computational Linguistics (TACL)*, 2014. 5, 15, 19, 20, 94

Haiqi Huang, Yueming Lu, Fangwei Zhang, and Songlin Sun. A multi-modal clustering method for web videos. In *ISCTCS*. 2013. 6

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *Proceedings of the 2014 ACM on Multimedia Conference*, 2014. 17, 27

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 77, 84, 85, 95, 96, 101, 105

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. 5

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015. 3, 5, 48, 94, 95

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. 1, 6

Chloé Kiddon, Luke S. Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 110

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 34, 43

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics (TACL)*, 2015. 3, 5, 48, 94, 95

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010. 15

A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision (IJCV)*, 50(2), 2002. 1, 6

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. *arXiv preprint arXiv:1611.06607*, 2016. 96, 105, 106, 107

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. *arXiv preprint arXiv:1705.00754*, 2017. 102, 103

Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, July 2013. 2, 6, 15, 19, 20, 100

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 2, 8, 15, 17, 18, 74

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. 5

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1, 5

Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012. 1

Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, UNC Chapel Hill, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. In *Transactions of the Association for Computational Linguistics (TACL)*, 2014. 3

Thomas K Landauer. *Latent semantic analysis*. Wiley Online Library, 2006. 107

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 8

Yong Jae Lee and Kristen Grauman. Predicting important objects for egocentric video summarization. *International Journal of Computer Vision (IJCV)*, 2015. 101, 107, 111

M.W. Lee, A. Hakeem, N. Haering, and S.C. Zhu. Save: A framework for semantic annotation of visual events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 1, 6

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 *Advances in Neural Information Processing Systems (NIPS)*, 2010. 7

S. Li, G. Kulkarni, T.L. Berg, A.C. Berg, and Y. Choi. Composing simple image descriptions using web-scale N-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL)*, 2011. 1, 5

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 5, 15, 19, 20, 44, 48, 50, 55, 94

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004. 85

Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2714–2721, 2013. 101, 107, 111

Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1942–1950, 2016. 75, 102

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 (NIPS)*, pages 1682–1690, 2014. 5

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2015. 99, 108

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. 3, 5, 48, 54, 94, 95

Junhua Mao, Xu Wei, 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 2533–2541, 2015. 99

Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proceedings of the Human Robot Interaction Conference (HRI)*, 2010. 5

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Advances in Neural Information Processing Systems (NIPS)*, 2013. 42, 55

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. 19

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL)*, 2016. 107

Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 29

Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1143–1151, 2011. 94

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. 6

Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 44, 75, 78, 97, 98

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. 20, 44, 85

Cesc C Park and Gunhee Kim. Expressing an image stream with a sequence of natural sentences. In *Advances in Neural Information Processing Systems (NIPS)*, pages 73–81, 2015. 96, 106

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 12:1532–1543, 2014. 42, 43, 50, 55

Yair Poleg, Chetan Arora, and Shmuel Peleg. Temporal segmentation of egocentric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2537–2544, 2014. 100, 101

Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 540–555, 2014. 77, 80, 81, 86, 87, 88, 89, 90, 100, 101, 103, 107, 111

Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. Multimodal video description.

In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1092–1096, 2016. 98, 110

Mengye Ren, Ryan Kiros, and R Zemel. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems (NIPS)*, volume 1, page 3, 2015. 5

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015. 81, 96, 101

Iain E Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004. 86

Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1

Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *German Conference on Pattern Recognition (GCPR)*, 2014. 101

Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. *German Conference on Pattern Recognition (GCPR)*, 2015. 34, 35, 100

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 27, 30, 31, 35, 44, 76, 83

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision (IJCV)*, 2017. 76, 83

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. ILSVRC Imagenet Large Scale Visual Recognition

Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 15, 17, 29, 48, 50, 56, 95

Shagan Sah, Sourabh Kulhare, Allison Gray, Subhashini Venugopalan, Emily Prud'Hommeaux, and Raymond Ptucha. Semantic text summarization of long videos. In *IEEE Winter Conference on Applications of Computer Vision (WACV), 2017*, pages 989–997, 2017. 107

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, Nov 1997. 78

Andrew Shin, Katsunori Ohnishi, and Tatsuya Harada. Beyond caption to narrative: Video captioning with multiple sentences. In *IEEE International Conference on Image Processing (ICIP)*, 2016. 102

Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016. 75, 102

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 26

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2014. 8, 29, 43, 48, 55, 77, 80

Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970, 2016. 75, 102

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *International Conference on Computer Vision (ICCV) Workshops*, 2012. 29

M. Sundermeyer, R. Schluter, and H. Ney. LSTM neural networks for language modeling. In *INTERSPEECH*, 2012. 39, 50

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014. 3, 8, 10, 13, 14, 15, 19

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 8, 32

Makarand Tapaswi, Martin BÃd'uml, and Rainer Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 99, 100, 108

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. 5

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 *International Conference on Computational Linguistics (COLING)*, 2014. 3, 7, 15, 19, 21, 22, 23, 32, 33, 99, 100

Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. 109

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. 27, 30, 31, 44, 76, 83

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 32

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. Context-aware captions from context-agnostic supervision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 106

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3, 26, 44, 75, 77, 108

Subhashini Venugopalan, Huijuan Xu, , Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2015. 3, 13, 20, 26, 29, 32, 33, 35, 75, 97

Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. Improving lstm-based video description with linguistic knowledge mined from text. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 4, 38, 54, 75

Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. Captioning images with diverse objects. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 48, 56

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015. 3, 5, 8, 13, 14, 15, 29, 40, 49, 75, 77, 94, 95, 97

Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, 2011. 7, 100

Shikui Wei, Yao Zhao, Zhenfeng Zhu, and Nan Liu. Multimodal fusion for video search reranking. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2010. 6

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 7, 100

Huijuan Xu, Subhashini Venugopalan, Vasili Ramanishka, Marcus Rohrbach, and Kate Saenko. A multi-scale multiple instance video description network. *ICCV Workshop on Closing the Loop between Vision and Language*, 2015. 98

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *IEEE International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015. 95, 96, 97

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. 19, 20, 21

Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011. 1, 99, 100

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. 1

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. *IEEE International Conference on Computer Vision (ICCV)*, 2015. 32, 33, 35, 37, 75, 77, 78, 97, 98, 108

Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016. 75, 102

Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from videos described with sentences. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013. 1, 6

Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. Visual madlibs: Fill in the blank image generation and question answering. *IEEE International Conference on Computer Vision (ICCV)*, 2015. 5

Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 78, 97, 98, 102, 106, 107

Wojciech Zaremba and Ilya Sutskever. Learning to execute. *Advances in Neural Information Processing Systems (NIPS)*, 2014. 10

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 2014. 15

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 766–782, 2016. 77, 86, 87, 88, 89, 90, 100, 101, 107

Yukun Zhu, Ryan Kiros, Rich 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. In *IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015. 99, 100, 108