Video description by learning to detect visual tags

A thesis submitted
in Partial Fulfillment of the Requirements
for the Degree of

Master of Technology

by

Rohit Gupta

to the
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
November, 2017

CERTIFICATE

It is certified that the work contained in the thesis titled **Video description by learning to detect visual tags**, by **Rohit Gupta**, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr Vinay Namboodiri
Department of Computer Science & Engineering
IIT Kanpur

November, 2017

ABSTRACT

Name of student: **Rohit Gupta** Roll no: **XXXXXXXXX**

Degree for which submitted: **Master of Technology**

Department: **Computer Science & Engineering**

Thesis title: **Video description by learning to detect visual tags**

Name of Thesis Supervisor: **Dr Vinay Namboodiri**

Month and year of thesis submission: **November, 2017**

A set of problems associating images and videos with a natural language descriptions, called Vision-to-Language (V2L) problems have attracted a lot of attention recently due to their potential to transform the paradigm of human-computer interactions and information retrieval. In particular, much progress has been made in the difficult task of video description using models which combine Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The state of the art models for this task typically do not try to explicitly detect visual semantic concepts, instead progressing from video to text via a black box model. We utilise techniques from object detection and action recognition research to explicitly recognize visual semantic tags and utilise Long Short Term Memory (LSTM) models for caption generation. We experiment with different architectures and evaluate our models on two open domain video datasets collected from YouTube, the Microsoft Research Video Description (MSVD) Corpus and Video-To-Text (MSR-VTT 2016) Corpus.

Acknowledgements

First and foremost I would like to extend my gratitude to Dr. Vinay P. Namboodiri, my thesis advisor, who was incredibly patient and an infinite fount of computer vision knowledge and practical research wisdom. I would also like to thank my parents who provided me their constant support. I am also thankful to my fellow students Samik Some, Debjeet Majumdar, Ayushman Sisodiya and others for helping me with my research. I also want to thank Dr. Amitabha Mukerjee and Dr. Gaurav Sharma for supervising my machine learning course projects and helping me hone my skills. Finally, I extend my gratitude to IIT Kanpur for providing me with the research facilities and administrative support required for my work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Vision-To-Language problems, such as the task of describing visual content have received a lot of interest lately from researchers. This is partly because making machines mimic such complex human behavior as understanding visual content is a an important goal for artificial intelligence research. Additionally video description has a wide variety of practical applications in the fields of information retrieval and human computer interaction.

Put simply, the task of automated Video Captioning entails training a model to describe events in a video using natural language. More formally, the task can be represented as modelling the probability $\boldsymbol{P}(\boldsymbol{w}|\boldsymbol{v})$, where $\boldsymbol{w} = \{w_1,...,w_t\}$ is a caption (a sequence of words) and $\boldsymbol{v} = \{f_1,...,f_k\}$ is a video (a sequence of frames). The task is illustrated by an example in Figure 1.1.

The task of video captioning can be thought of as a generalization of the task of image captioning. However the image captioning task [1] [2] has a fixed size input, a single frame, whereas a video captioning model has to work with variable length videos which can have both local and global structure. Even very short video clips (5 seconds) can have complex global interactions between multiple subjects involving multiple objects with complex attributes. [3]

In this work we tackle the video captioning problem by splitting it into 2 sub-tasks. First we detect visual semantic tags in the video clip and then using these predicted tags we generate captions for the video. Symbolically, this split can be represented

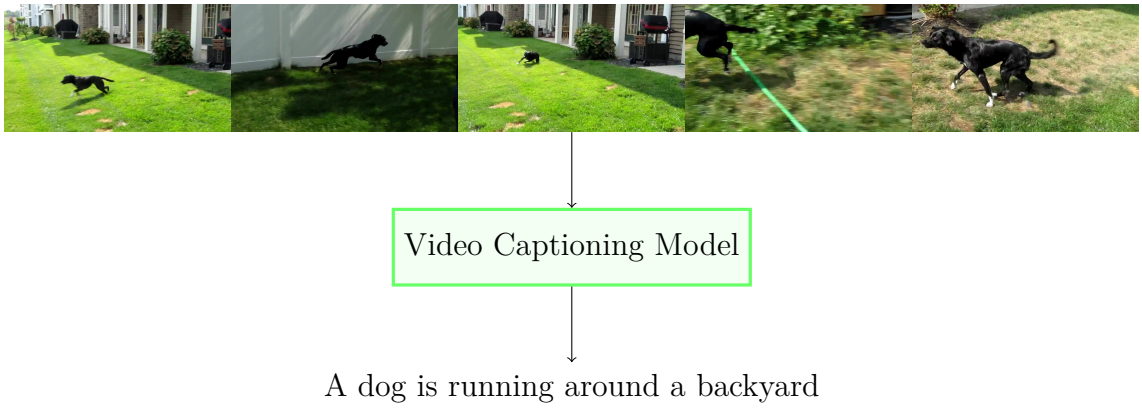


Figure 1.1: Video Captioning Task: Illustrated by example from MSVD dataset

by $\boldsymbol{P}(\boldsymbol{w}|\boldsymbol{v}) = \sum_h \boldsymbol{P}_\theta\ (\boldsymbol{w}|h)\boldsymbol{P}(h|\boldsymbol{v})$ where w and v have the same meaning as before and h denotes visual concepts or "tags". We have explicitly factorized the captioning model $\boldsymbol{P}(\boldsymbol{w}|\boldsymbol{v})$ into two parts, $\boldsymbol{P}(\boldsymbol{w}|h)$, which is a language model conditioned on tags, and $\boldsymbol{P}(h|\boldsymbol{v})$, which is a tag recognition model on visual inputs. This scheme is similar to the one followed by [4]

The rest of this thesis is organized as follows: In Chapter 2 we talk about some recent research related to or relevant to our problem. Chapter 3 discusses some background techniques, concepts and models required to understand our approach to the problem. Chapter 4 explains in detail our method to solving the problem and the models used. In Chapter 5 we present the design and results of various experiments performed and the datasets used to evaluate our method against pre-existing methods. Finally we conclude with some suggestions for future work in Chapter 6.

Chapter 2

Related work

2.1 Image Captioning

The task of image captioning connects two major artificial intelligence fields: computer vision and natural language processing. Typical Deep Learning approaches for this problem utilize an Encoder-Decoder framework. In the Encoding step, the source image is encoded into an embedding vector. This is followed by the decoding step, in which a caption is generated by decoding the embedding vector. Variations on this framework change how to encode the source information image and how to decode the image embeddings.The first such model was presented in [1] where a CNN is used as an image encoder and an LSTM as the caption decoder. In [5] a similar model is used for image captioning, however a additional alignment model using CNN over image regions and RNN over sentences is used to compute Visual-Semantic Alignments. Yet another class of image captioning models derivative of the CNN-RNN approach are attention based models such as [6] and [2]. These attention based methods allow their model to attend to any visual parts of the input image.

2.2 Video Classification

Current state of the art results in video classification are achieved in [7] using temporal pooling and LSTM architectures to combine image information across a

video over longer time periods. Another frequently used method for this task is to Train a deep 3D convolutional network on a large video dataset annotated with objects, actions, scenes [8]. The use of 3D CNNs for video classification was first proposed in [9] which relied on a 2-stream network which operated on a context (resized) and a fovea (center cropped) stream. [10] also present a Two-stream CNN architecture for video classification, however in this case the 2 streams used are a single RGB frame and multi-frame optical flow.

2.3 Video Captioning

The earliest video captioning systems relied on a rule-based systems for describing ego-centered activities with natural language. [11] These were restricted to limited settings where sentences were generated by filling in predefined templates with recognized objects and actions. [12, 13] Some methods went beyond rule engineering by training statistical models for lexical entries. [14, 15, 16, 17] This allows video captioning on open domain datasets but generation performance of these methods is usually low. Recently models inspired by the Sequence-to-Sequence approach pioneered by Neural Machine translation [18] have achieved the best performance on the video captioning task. These models could use a 3d-CNN encoder followed by a LSTM decoder [3], a frame by frame LSTM encoder and sentence decoder [19] or an average pooling encoder followed by an LSTM decoder. [20] Building on these methods [21, 22] achieve state of the art results on various video captioning datasets using hierarchical recurrent networks.

2.4 Video Captioning With Visual Tags

In [4] a trainable oracle model is proposed that helps measure maximum achievable performance when a high quality visual feature detector is available. It is found that the state of the art video captioning models significantly underperform the oracle, hence suggesting that visual detectors have a lot of room for improvement. In [23] the

authors train visual classifiers from the weak annotations of the sentence descriptions which distinguish verbs, objects, and places. Based on the results of these visual classifiers they learn how to generate a description using an LSTM language model. Temporal, motion, and semantic features(tags) are used in [24] to achieve state of the art captioning results.

Chapter 3

Background

3.1 Neural Networks and Deep Learning

Neural Networks are machine learning models structured as layers stacked one after the other. A neural network learns successive layers of representations. The term "neural network" is a reference to the fact that some of the central concepts in deep learning were developed in part by drawing inspiration from the scientific understanding of human/animal brains. A simple neural network is illustrated in Figure 3.1

In deep learning models, Model input data and target data are both first vectorized into separate input and target vector spaces. Each layer in a deep learning model performs a simple geometric transformation on its input data to compute the output. Together, the stacked layers of the model form a complex geometric transformation, broken down into a series of simple ones. [25, 26] This complex transformation attempts to maps the input space to the target space controlled by the weights of the layers, which are iteratively updated based on the accuracy of the model's prediction on the current batch of training data. A key characteristic of this geometric transformation is that it must be differentiable, which is required in order to train its parameters via gradient descent. This imposes an additional requirement that the transformation from inputs to outputs must be smooth and continuous. [27]

Even though the earliest Neural Networks were designed in the 1960s [28], the first

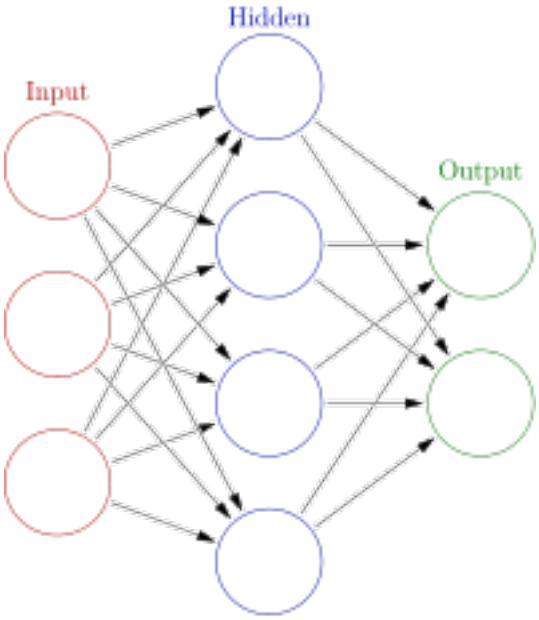


Figure 3.1: A toy example of an Artificial Neural Network (By Aphex34 (Own work)) licensed under CC BY-SA 4.0, via Wikimedia Commons

successful commercial application of neural nets came in 1989 from Yann LeCun who combined the earlier ideas of convolutional neural networks and backpropagation, and applied them to the problem of handwritten digits classification [29]. The ”LeNet” model so developed was used by the US Postal Service to automate the reading of ZIP codes on mail. [30]

3.2 Convolutional Neural Networks

The very first successful practical application of neural nets as mentioned above was using a convolutional neural networks applied to the problem of handwritten digits classification [29]. ConvNet architectures make the explicit assumption that its inputs are images, which vastly reduces the number of parameters in the network and helps exploit the structure present in images. Unlike a simple Neural Network, the neurons in the layers of a ConvNet are arranged in 3 dimensions: width, height, depth. The fundamental building blocks of a ConvNet are Convolutional Layers, Pooling Layers, and Fully-Connected Layers. [31]

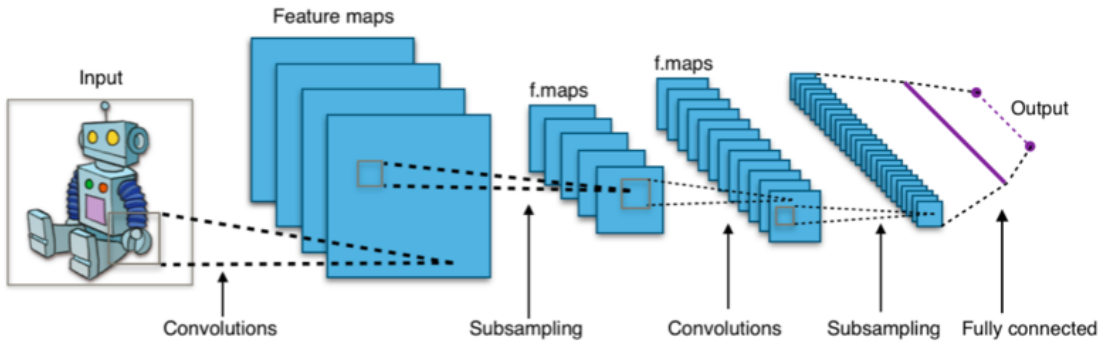


Figure 3.2: A simple Convolutional Neural Network (By Aphex34 (Own work)) licensed under CC BY-SA 4.0, via Wikimedia Commons

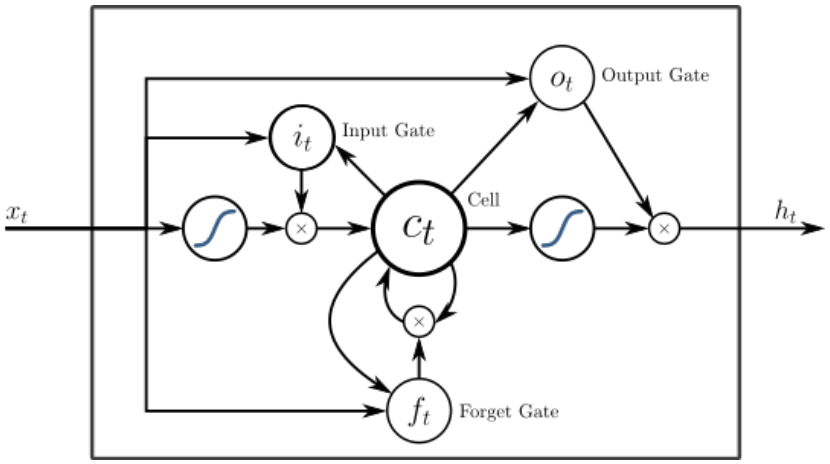


Figure 3.3: Structure of an LSTM Cell (By Alex Graves et al (Own work)) licensed under CC BY-SA 4.0, via Wikimedia Commons

3.3 Recurrent Neural Networks

Recurrent neural networks are the architecture of neural network to use for learning a model on sequences and list data. RNNs have been succesful at a variety of machine learning problems involving sequence data: speech recognition, language modeling, translation, image captioning and so on [32] Long Short Term Memory networks (LSTMs) are a special type of RNN, capable of learning long-term dependencies. [33]

Chapter 4

Method

4.1 Approach

As explained earlier we split the problem of video captioning into two simpler tasks:

- Predicting visual tags given a video clip.
- Generating captions for the video clip using the predicted tags.

In the following sections we describe each step in detail.

4.1.1 Tag Prediction

Before we go further we must describe what is meant by tags. Simply put tags are specific visual attributes of the video clips, such as objects, actions and attributes present in them. It is represented in a bag of words form (a binary vector) over a pre-defined vocabulary of tags. As such in this step, given a video clip we would like to predict all tags in the vocabulary found in the clip to it. The datasets we use do not provide us with ground-truth tag sets, but only videos and their captions. We generate a set of ground-truth tags for each video from its caption by removing stopwords from it followed by classifying the tags into categories based on parts of speech. We use the Stanford Log-linear Part-Of-Speech Tagger [34] to tag each caption and then group the tags into 3 categories following the scheme presented in [4]:

- **Entity Tags:** Words tagged with “NN”, “NNP”, “NNPS” ,“NNS”, “PRP”
- **Action Tag:** Words tagged with “VB”, “VBD”, “VBG”, “VBN”, “VBP”, “VBZ”
- **Attribute Tag:** Words tagged with “JJ”, “JJR”, “JJS”

The top ten visual tags in each category (and their frequency) in the MSVD Dataset after this process has been carried out are:

Entities: man (22885), woman (10211), person (4493), girl (3156), dog (3086), cat (3063), boy (2685), guitar (2300), baby (2144), water (2065)

Actions: playing (7195), cutting (2662), riding (2643), dancing (2385), slicing (2166), walking (1690), eating (1680), running (1311), doing (1256), talking (1164)

Attributes: small (963), young (780), little (739), other (682), large (410), white (374), green (272), several (258), garlic (229), black (189)

We try two different techniques for predicting visual tags:

Fully-connected Network (FCN)

We use a simple fully connected feedforward (FCN) to predict the tags directly from the averaged ResNet-50 representation of the video clip (L2-normalized average over frames). This is a multilabel learning scenario and we use binary cross-entropy loss over sigmoid outputs to predict the tags. We use grid search over hyperparameters to find optimal number of layers, layer size and regularization strength.

Recurrent Network for Video Classification (LSTM)

Following [7] we also tried independently processing each frame of each video using a CNN (ResNet-50 pre-trained on Imagenet) and then training an LSTM to use the frame-level features over the entire video to label it. Grid search was also done to find the optimal number of layers and hidden state size for the LSTM stack.

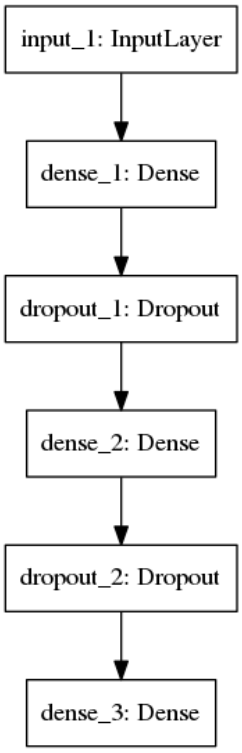


Figure 4.1: Simple feedforward network for tag detection

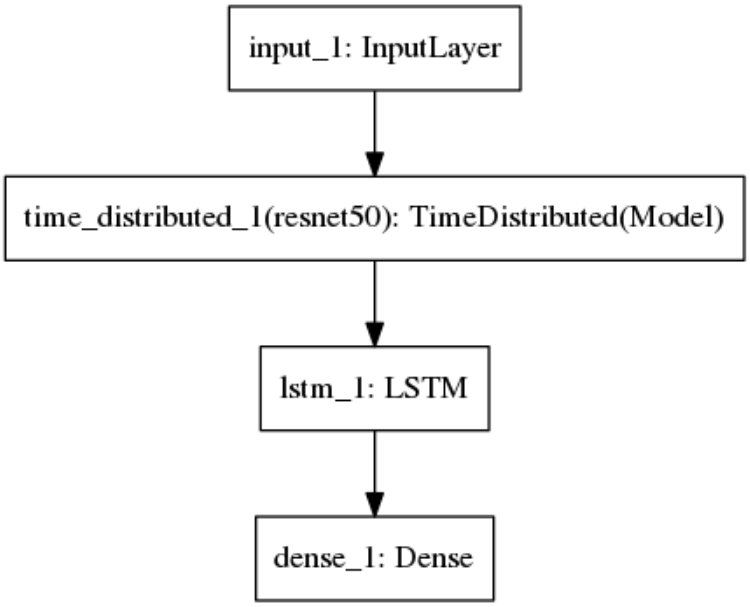


Figure 4.2: Frame Level LSTM Network for tag detection

4.1.2 Caption Generation

Once we have predicted a set of tags for a video clip we can use it for the caption generation step. We use a conditional language model for generating captions. An LSTM is used as the language model [32] as shown in Figure 4.3. As input to the LSTM we provide the ResNet-50 features of the video clips as well as predicted tags. We also feed back the partial caption generated before this time step to the LSTM at every time-step. To start prediction the network is given a special start token and prediction continues until it reaches a predefined maximum number of words.

Symbolically, the task of conditional language modelling can be represented as learning to model the probability $\boldsymbol{P}(w_{n+1}|w_n, \boldsymbol{h})$, where $\boldsymbol{h}$ are the visual tags detected in the video and w_n is the nth word of the caption.

To improve the accuracy of the language modelling, we added two additional features over the vanilla LSTM: Teacher Forcing [35] and Beam Search [1].

Teacher forcing: While training we feed back the true partial caption rather than the predicted one.

Beam Search: Beam Search involves iteratively considering the top K sentences up to timestep t as candidates to generate sentences of size t + 1, and keeping only the best K of them. We use beam search with beam size K=3 to generate captions.

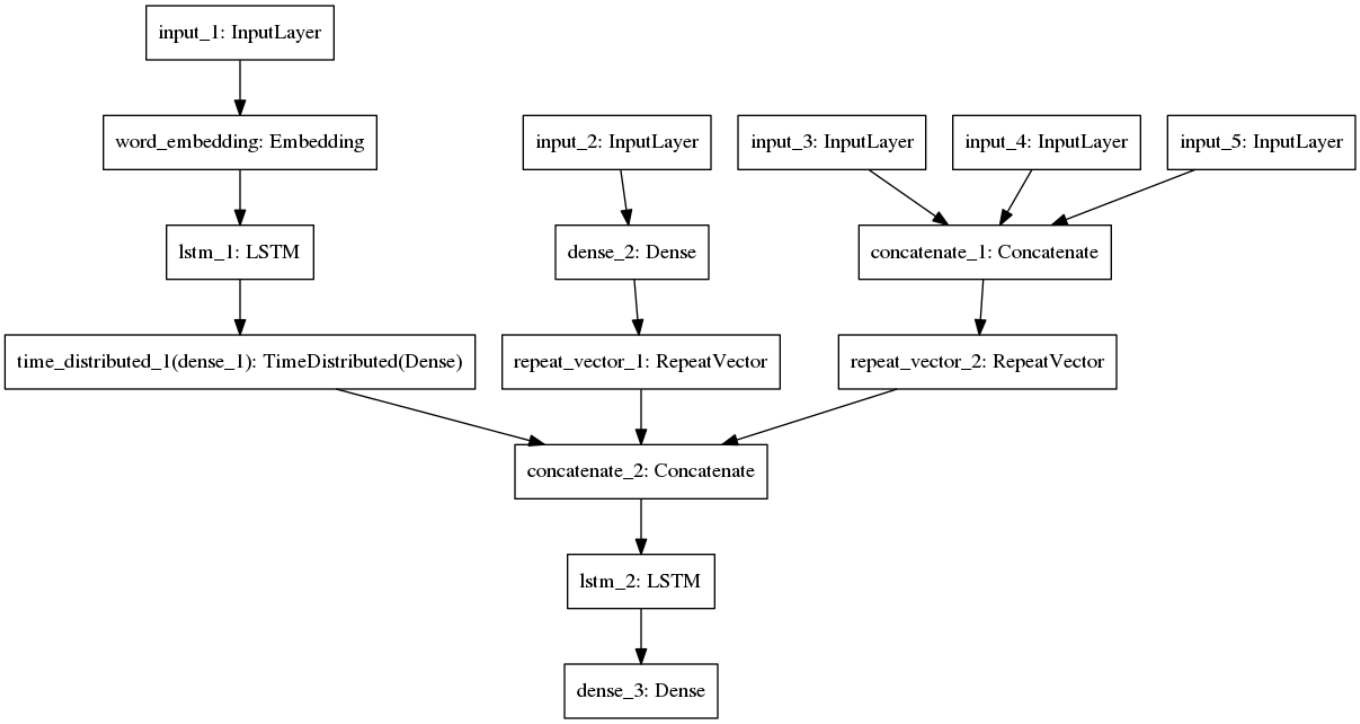


Figure 4.3: Conditional Language Model for caption generation. Input 1 is the partial caption generated till timestep t-1, input 2 is the averaged ResNet50 features over the frames of the video and inputs 3 through 5 are the predicted visual tags. Output dense 3 is the predicted word at timestep t

Chapter 5

Experiments and Analysis

5.1 Datasets

We make use of two different datasets collected from YouTube by Microsoft Research: MSVD [36] and MSR-VTT [37]

- The MSVD dataset consists of 1970 video clips obtained from YouTube. In contrast to the previous dataset, MSVD provides multiple captions for each video clip. There are about 16 verified English captions for each video with a total of 25850 captions. This increases further to 85550 captions if we use the unverified captions as well.

- The MSR-VTT dataset consists of 10000 video clips obtained from YouTube. Like MSVD, MSR-VTT also provides multiple captions for each video clip. There are about 20 English captions for each video with a total of 200,000 captions. The dataset contains YouTube clips in 20 categories. The number of videos in each category is presented in Figure 5.1

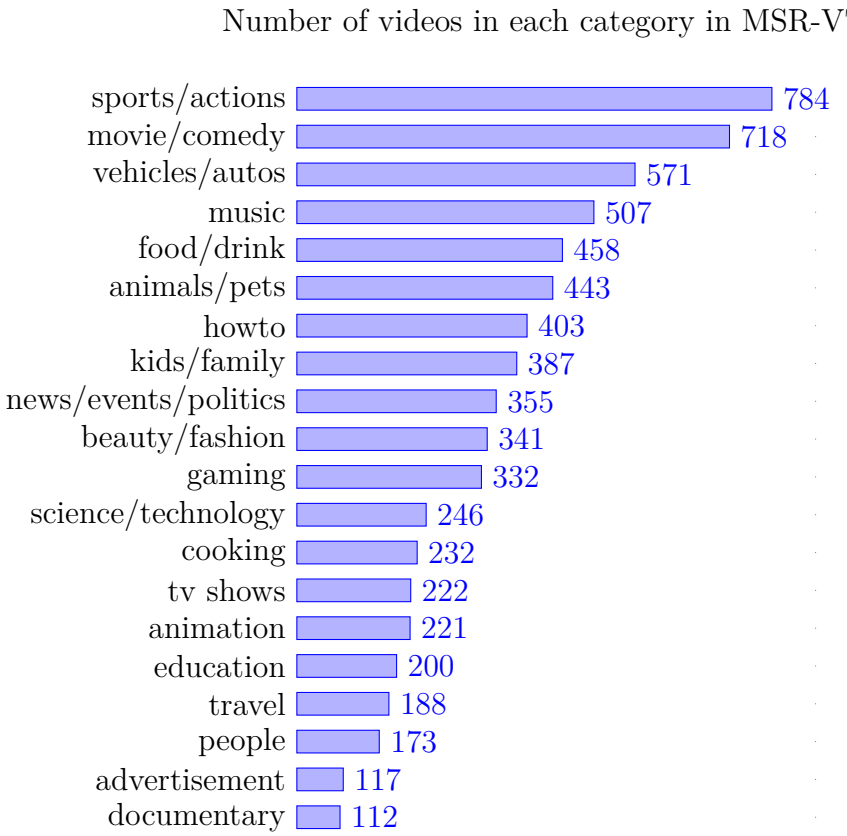


Figure 5.1: Number of videos in each category in MSR-VTT

5.2 Results

5.2.1 Quantitative Results

We present our results for both sub-tasks and for the overall captioning tasks in this section. The performance of the two different tag prediction models is documented in table 5.1. We report Micro Averaged Precision (μAP) as the labels are thoroughly unbalanced and biased towards the most common labels. To calculate (μAP) we sum up the individual true positives, false positives, and false negatives of the system for all labels across all tag types: $\mu AP = \Sigma TP/(\Sigma TP + \Sigma FP)$

In 5.2 we report the confusion matrix metrics for the FCN tag prediction model for a probability threshold of 1.5%. It is observed that False positives are actually higher than True positives, as a result the precision of the predicted tags is relatively low. However the tag prediction model recalls about 2/3rds of the correct tags.

Model	FCN	LSTM
MSVD Dataset		
Tag Type		
Entity	0.51	0.51
Action	0.40	0.40
Attribute	0.25	0.23
MSR-VTT Dataset		
Tag Type		
Entity	0.34	0.37
Action	0.43	0.40
Attribute	0.20	0.17

Table 5.1: Micro Averaged Precision (μAP) of Tag Prediction Results

Confusion Matrix at probability threshold 0.015		
-	Condition Positive	Condition Negative
Predicted Condition Positive	2.53%	3.25%
Predicted Condition Negative	1.45%	92.77%
Precision	43.79%	
Recall	63.54%	

Table 5.2: Confusion Matrix for FCN Tag Prediction results

This suggests that pruning the set of predicted tags could lead to improvements. Preliminary explorations in this direction are promising and we are able to match S2VT-RGB performance after pruning tags below a certain threshold.

In table 5.3 we provide METEOR [38] and CIDEr [39] scores for the captions generated on both datasets. We observe that there is a significant gap between caption generation results using ground truth tags and using predicted tags. This suggests that overall performance of the model can be significantly improved by improving the quality of the tag prediction model.

In table 5.4 we compare our results with some state-of-the art models. Our model does not use motion information (optical flow), unlike the top ranked models, as a result the best baseline comparison for our model is the end-to-end S2VT model which also relies on purely RGB video information. We achieve near parity with the S2VT-RGB model.

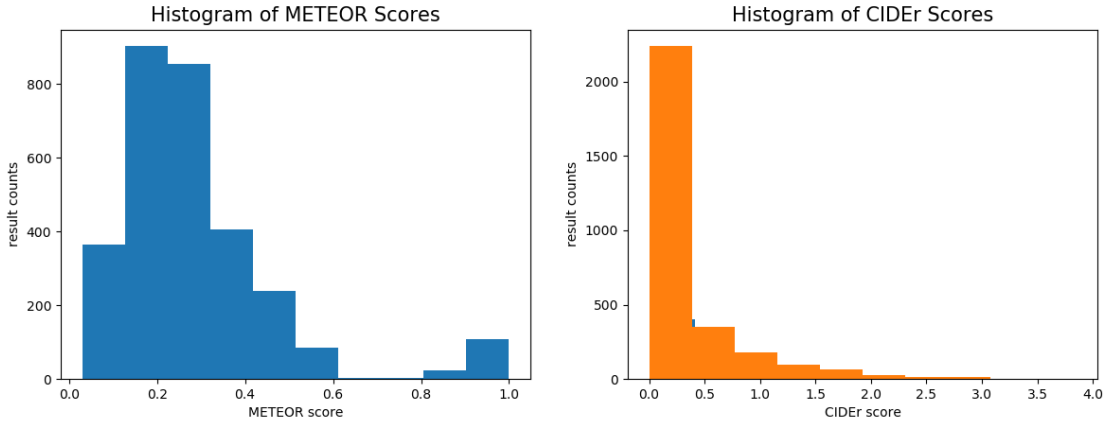
In Figures 5.2a and 5.2b we provide the distribution of CIDEr and METEOR

Tag Type	METEOR	CIDEr
MSVD Dataset		
Ground-truth	0.404	1.084
FCN predicted	0.285	0.508
LSTM predicted	0.277	0.496
MSR-VTT Dataset		
Ground-truth	0.306	0.549
FCN predicted	0.249	0.321
LSTM predicted	0.251	0.335

Table 5.3: Caption Generation Results

Model	METEOR (in %)
MSVD Dataset	
S2VT	29.8
p-RNN	32.6
HRNE (Pan et al)	33.1
Multi-Faceted Attention (Long et al)	33.4
S2VT-RGB	29.2
Ours	28.5
Ours w/tag pruning (preliminary investigations)	29.0
Ours w/groundtruth tags	40.4
MSR-VTT Dataset	
v2t_navigator (Winner, MSR-VTT 2016)	28.2
VideoLAB (Runner-Up, MSR-VTT 2016)	27.7
Multi-Faceted Attention (Long et al)	26.9
Ours	25.1
Ours w/groundtruth tags	30.6

Table 5.4: Ours vs State of the Art Models



(a) METEOR Scores **(b)** CIDEr Scores

Figure 5.2: Distribution of METEOR and CIDEr Scores of predicted captions for MSR-VTT dataset

scores over the captions in the test set to display the variation in generated sentence quality. As we can observe from the distribution of the METEOR and CIDEr scores some sentences generated are of very high quality while most others are very close to the median score. A small portion of the test videos: 200 videos ($<$ 10% of the dataset) have very high quality captions generated. Figures 5.3 and 5.4 are from this group, while Figure 5.5 is close to the median METEOR score.

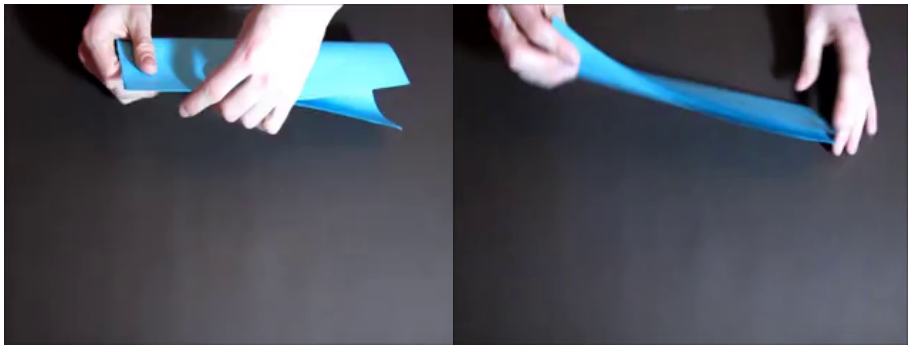


Figure 5.3: Generated Caption: a person is folding a piece of paper
Reference Captions: a guy folding paper
a man is folding a piece of paper
a man is folding the paper
a person folding a piece of paper on a table
a person folds up paper to make a paper plane
a person is folding a blue paper in half
a person is folding a piece of paper in half
a person is folding an 8x11 piece of blue paper
a person is folding origami
a person is folding paper

5.2.2 Qualitative Results

In this section we present a few sample captions generated by our model along with keyframes from their corresponding video clips in Figures 5.3, 5.4 and 5.5. The sentences broadly convey the contents of the video covering both objects and actions. However in some cases, some visual tags are incorrectly identified, e.g. ”man” instead of ”woman” in Figure 5.5. Another feature that stands out is that most generated captions use only a small portion of the total vocabulary, not generating more uncommon context appropriate words. Both these shortcomings point towards avenues for further improvement.

Figure 5.4: Generated Caption: a person is playing a video game
Reference Captions: a guy is playing a car game
a man is commentating while playing a video game
a man is crashing into other cars in a video game
a man makes commentary over a driving game
a person is playing a computer online game
a person is playing a video game
playing a driving video game
a youtuber playing mad max fury road
gameplay footage of someone playing a game
person playing a video game and talking about it

Figure 5.5: Generated Caption: a man is talking about a car
Reference Captions: a car is driving
a foreign lady is speaking about a a car
a lady is talking in front of a car
a spanish commercial for a honda civic with a woman in a flowery top presenting
a spanish speaking woman presenting a car
a woman discussing a car
a woman is driving a car and talking about that car
a woman is talking about a car
a woman is talking while standing in front of a black car

Chapter 6

Conclusions

The main aim of our work was to demonstrate that the task of video description could be formulated as a combination of visual concept detection (Action/Object/Attribute Recognition) and conditional language modelling. We demonstrated this successfully, while achieving close to state of the art performance.

6.1 Scope for further work

Since we have successfully split the task of video captioning, this means the rapid advances in object detection, action recognition and language modelling could help us get near automatic improvements in video description itself. Additionally, the impact of integrating newer data modalities like motion information (optical flow) and audio is an interesting avenue for exploration. Yet another avenue that could possibly be taken up is using additional unlabelled or semi-labelled data to improve our tag prediction models.

References

[1] Oriol Vinyals et al. "Show and tell: A neural image caption generator". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.

[2] Kelvin Xu et al. "Show, attend and tell: Neural image caption generation with visual attention". In: *International Conference on Machine Learning*. 2015, pp. 2048–2057.

[3] Li Yao et al. "Describing videos by exploiting temporal structure". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4507–4515.

[4] Li Yao et al. "Oracle performance for visual captioning". In: *CoRR* abs/1511.04590 (2015). URL: http://arxiv.org/abs/1511.04590.

[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*. 2015, pp. 3128–3137.

[6] Quanzeng You et al. "Image captioning with semantic attention". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4651–4659.

[7] Joe Yue-Hei Ng et al. "Beyond short snippets: Deep networks for video classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4694–4702.

[8] Du Tran et al. "Learning spatiotemporal features with 3d convolutional networks". In: *arXiv preprint arXiv:1412.0767* (2014).

[9] Andrej Karpathy et al. "Large-scale Video Classification with Convolutional Neural Networks". In: *CVPR*. 2014.

[10] Karen Simonyan and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos". In: *Advances in neural information processing systems*. 2014, pp. 568–576.

[11] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. "Natural language description of human activities from video images based on concept hierarchy of actions". In: *International Journal of Computer Vision* 50.2 (2002), pp. 171–184.

[12] Mun Wai Lee et al. "Save: A framework for semantic annotation of visual events". In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE. 2008, pp. 1–8.

[13] Andrei Barbu et al. “Video in sentences out”. In: *arXiv preprint arXiv:1204.2742* (2012).

[14] Sergio Guadarrama et al. “YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-Shot Recognition”. In: *The IEEE International Conference on Computer Vision (ICCV)*. Dec. 2013.

[15] Niveda Krishnamoorthy et al. “Generating Natural-Language Video Descriptions Using Text-Mined Knowledge.” In: 2013.

[16] Huijuan Xu et al. “A multi-scale multiple instance video description network”. In: *arXiv preprint arXiv:1505.05914* (2015).

[17] Chen Sun and Ram Nevatia. “Semantic aware video transcription using random forest classifiers”. In: *European Conference on Computer Vision*. Springer. 2014, pp. 772–786.

[18] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[19] Subhashini Venugopalan et al. “Sequence to Sequence - Video to Text”. In: *CoRR* abs/1505.00487 (2015). URL: http://arxiv.org/abs/1505.00487.

[20] Subhashini Venugopalan et al. “Translating videos to natural language using deep recurrent neural networks”. In: *arXiv preprint arXiv:1412.4729* (2014).

[21] Pingbo Pan et al. “Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning”. In: *CoRR* abs/1511.03476 (2015). URL: http://arxiv.org/abs/1511.03476.

[22] Haonan Yu et al. “Video Paragraph Captioning using Hierarchical Recurrent Neural Networks”. In: *CoRR* abs/1510.07712 (2015). URL: http://arxiv.org/abs/1510.07712.

[23] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. “The Long-Short Story of Movie Description”. In: *Proceedings of the German Confeence on Pattern Recognition (GCPR)*. Vol. 9358. Lecture Notes in Computer Science. Oral, Honorable Mention prize. Springer International Publishing. Springer International Publishing, 2015, pp. 209–221. ISBN: 978-3-319-24946-9. DOI: 10.1007/978-3-319-24947-6_17. URL: http://arxiv.org/abs/1506.01698.

[24] Xiang Long, Chuang Gan, and Gerard de Melo. “Video captioning with multi-faceted attention”. In: *arXiv preprint arXiv:1612.00234* (2016).

[25] Chris Olah. *Neural Networks, Manifolds, and Topology*. URL: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/.

[26] F. Chollet. *Deep Learning with Python*. Reference copy was Manning Early Access Program Version 4. Manning Publications Company, 2017. ISBN: 9781617294433. URL: https://www.manning.com/books/deep-learning-with-python.

[27] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[28] Frank Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[29] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[30] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[32] Andrej Karpathy. *The Unreasonable Effectiveness of Recurrent Neural Networks*. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[33] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[34] The Stanford Natural Language Processing Group. *Stanford Log-linear Part-Of-Speech Tagger*. URL: https://nlp.stanford.edu/software/tagger.html.

[35] Samy Bengio et al. “Scheduled sampling for sequence prediction with recurrent neural networks”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 1171–1179.

[36] *Microsoft Research Video Description Corpus*. http://www.cs.utexas.edu/users/ml/clamp/videoDescription/. Accessed: 2017-07-30.

[37] *Microsoft Research Video To Language challenge 2016 Dataset*. http://ms-multimedia-challenge.com/2016/dataset. Accessed: 2017-09-05.

[38] Michael Denkowski and Alon Lavie. “Meteor Universal: Language Specific Translation Evaluation for Any Target Language”. In: *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*. 2014.

[39] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. “CIDEr: Consensus-Based Image Description Evaluation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.