Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Aditya Surikuchi

Visual Storytelling:

Captioning of Image Sequences

Master's Thesis
Espoo, November 25, 2019

Supervisor: Jorma Laaksonen D.Sc. (Tech.), Aalto University
Advisor: Jorma Laaksonen D.Sc. (Tech.), Aalto University

Aalto University
School of Science

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Aditya Surikuchi		
Title:	Visual Storytelling: Captioning of Image Sequences		
Date:	November 25, 2019	**Pages:**	78
Major:	Machine Learning, Data Science and Artificial Intelligence	**Code:**	SCI3044
Supervisor:	Jorma Laaksonen D.Sc. (Tech.), Aalto University		
Advisor:	Jorma Laaksonen D.Sc. (Tech.), Aalto University		

In the space of automated captioning, the task of visual storytelling is one dimension. Given sequences of images as inputs, visual storytelling (VIST) is about automatically generating textual narratives as outputs. Automatically producing stories for an order of pictures or video frames have several potential applications in diverse domains ranging from multimedia consumption to autonomous systems. The task has evolved over recent years and is moving into adolescence. The availability of a dedicated VIST dataset for the task has mainstreamed research for visual storytelling and related sub-tasks.

This thesis work systematically reports the developments of standard captioning as a parent task with accompanying facets such as dense captioning, and gradually delves into the domain of visual storytelling. Existing models proposed for VIST are described by examining respective characteristics and scope. All the methods for VIST adapt from the typical encoder-decoder style design, owing to its success in addressing the standard image captioning task. Several subtle differences in the underlying intentions of these methods for approaching the VIST are subsequently summarized.

Additionally, alternate perspectives around the existing approaches are explored by re-modeling and modifying their learning mechanisms. Experiments with different objective functions are reported with subjective comparisons and relevant results. Eventually, the sub-field of character relationships within storytelling is studied and a novel idea called character-centric storytelling is proposed to account for prospective characters in the extent of data modalities.

Keywords:	captioning, visual storytelling, sequence modeling, natural language processing, computer vision, semantic relationships, deep reinforcement learning
Language:	English

Acknowledgments

I would like to express my utmost gratitude to my supervisor and advisor, Jorma Laaksonen D.Sc. (Tech.), for nurturing my thoughts, laying my foundations and providing moral support.

I am thankful to my university and fellow Aalto CBIR research group mates: Arturs Polis, Julius Wang, Hamed R. Tavakoli, Phu Pham, Héctor Laria Mantecón, Selen Pehlivan Tort and Mats Sjöberg for improving my research acumen.

This work would not have been possible without the funding from MeMAD.

I am grateful to the janitorial staff, for maintaining my office space and the coffee machine.

Finally, I am indebted to my friends, for helping me cope with the drastic cultural and climatic differences, and to my parents, for always believing in my thoughts and endeavors.

Espoo, November 25, 2019

Aditya Surikuchi

Abbreviations and Acronyms

AI	Artificial Intelligence
BLEU	Bilingual Evaluation Understudy
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
EOS	End Of Sentence
GAN	Generative Adversarial Networks
GIF	Graphics Interchange Format
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
MDP	Markov Decision Process
METEOR	Metric for Evaluation of Translation with Explicit Ordering
ML	Machine Learning
MLE	Maximum Likelihood Estimation
NAACL	North American Chapter of the Association for Computational Linguistics
NLP	Natural Language Processing
RCNN	Region Convolutional Neural Network
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SCN	Semantic Compositional Network
SCST	Self-Critical Sequence Training
VIST	Visual Storytelling

Contents

Chapter 1

Introduction

1.1 Problem statement

Given a sequence of images as input, the visual storytelling task is about building a model that can generate a coherent textual narrative as output. An image sequence would typically be a group of images portraying an event or an episode. The output story could conceivably be up to fifty words long, with an average of ten words per image in the input sequence. The topic grew mainstream with advancements in deep-learning research achieving state-of-the-art performance for standard image captioning.

The first and only curated dataset for visual storytelling task, to date, is the VIST dataset, which is detailed under Section 2.3 and released by the work that popularized visual storytelling [22]. All the work that has followed has heavily relied on using the same dataset and proposed architectures that are pseudo dependent on the composition of the data. In this setting, coherency is a judgmental assessment of the style of translating visual semantics into language format.

Nevertheless, because sharing and maintaining context across sentences is the primary objective of successful visual storytelling and approaches for achieving it vary, several associated facets of the task remain subjective, including coherency. The balance between incorporating relevant details and creative abstractness coherent to the data is challenging. Moreover, other complications include accounting for ground-truth bias and sentence-length. These inherent difficulties and directions towards handling them will be discussed in this thesis.

1.2 Motivation

Computer vision, together with text and language processing, is successfully enabling new possibilities in many disciplines. With a significant surge in the availability of multimedia content everywhere, large-scale annotated data is becoming a reality. Automated generation of appropriate textual descriptions for images and videos is called automatic visual description. Some direct applications of these visual descriptions are understanding of images on social media platforms for better recommendations and captioning of videos on broadcasting mediums for the hearing or visually impaired. Other ancillary applications include describing signs and symbols in different levels of detail for the interpretability of robots in autonomous systems.

Although many works [35] have addressed the standard image and video captioning problems, the specific task of visual storytelling is a relatively new facet. Often more than not, the standard captioning models fail to interpret the non-obvious structure in the visual input. They do not account for different moments within the image or across a given sequence of images. These shortcomings of existing captioning methods form the motivation for generating a coherent narrative for a sequence of images or video frames comprising of relevant, subjective, and abstract information.

The VIST models are expected to generate stories with a balance between creativity and actuality of the data without loss of critical semantics. Ideally, such models learn the space between modalities accounting for the overall value of knowledge from both the visual data and the human-annotated textual interpretation.

1.3 Structure of the Thesis

This thesis intends to review the evolution of visual storytelling from the task of standard visual description. The contemporary state of the topic is subsequently explained with a discussion about possible sub-domains and suggestions towards future implications. Existing models and architectures are compared against each other illustrating respective leaps and shortcomings. Alternate perspectives and impact of remodeling some existing designs are discussed. A study on character relationships with a novel approach for solving one of the fundamental challenges of assuring the presence of prospective characters in the generated narratives is also proposed.

Structural outline of this thesis is as follows:

Chapter 2 discusses the relevant background, extensively covering the realms of image, video, and sequence captioning, thereby gradually introducing visual storytelling.

Chapter 3 compares the existing models proposed for visual storytelling by explaining the intentions behind, examining model behaviors, and reports respective implementations with results.

Chapter 4 describes modifications in learning mechanisms and remodeling of some existing methods and demonstrates resulting implications.

Chapter 5 discusses the difficulty of assuring prospective characters in the generated narratives and proposes a novel approach.

Chapter 6 summarizes and discusses data sources, model designs, experimental setups, and justifies the outputs of the models.

Chapter 7 concludes by reviewing the potential extent of the topic of research and motivates plausible future directions.

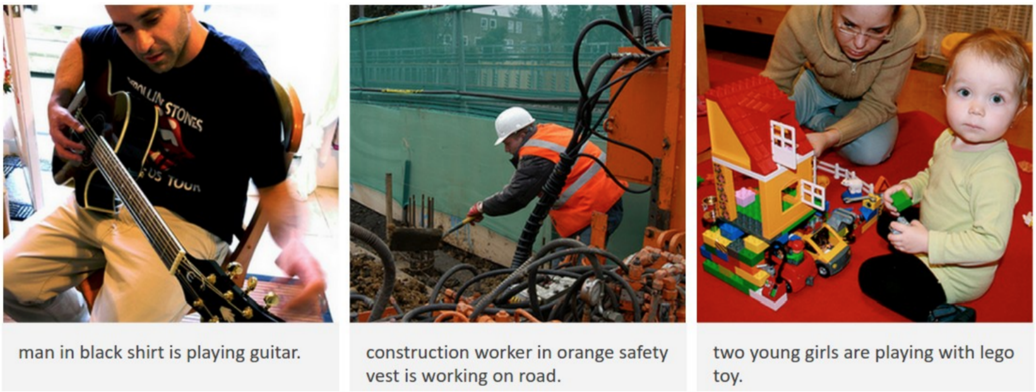
Chapter 2

Background

Visual storytelling is an extensively derived topic. It belongs to the family of captioning with connections to various other tasks. Automatically generated descriptions can assist people with visual or hearing impairment to perceive multimedia content. Real-time closed captioning on social media and broadcasting platforms solve the problems of language barriers and improve outreach. To fully understand how the research field arrived at VIST as a task, it is essential to study into respective parent and sibling domains. This chapter details each of the related topics and respective motivations behind them. A timeline view on the evolution of the space between vision and language and, thereby, visual storytelling is provided.

2.1 Image captioning

Image captioning is a task of automatically producing textual descriptions for given visual data as shown in Figure 2.1. The conception of captioning as a task is often primarily traced back to success in the field of visual object detection [32]. It can be viewed as a natural quest to progress towards describing the overall image or frame once the objects and entities are well tagged. Along with the novelty dimension, it could have been the influence of numerous other motivations that propelled this task into mainstream focus. Humans tend to perceive and learn visually, but communicate and share through text, language. This is rather evident considering many aspects of life from advertisements, social media to multimedia platforms. Therefore, the emphasis on language processing and understanding grew with the advent of deep language models utilizing neural networks [6]. Various improvements have followed, enhancing both the visual detection and text processing areas individually [40], [17]. Audio or speech is regarded arguably as a form of text



man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

Figure 2.1: Automatically generated captions by an AI model [25].

referring to the language modeling aspect in all standard speech processing and recognition pipelines.

All the motivations led to the initial work by Kiros et al [28] on image captioning using deep learning. The proposal elucidates the theme of multi-modalities and explores relationships between them. It presents a neural language model for generating textual output, conditioned on images without any predefined syntactic structures and templates that were relied upon by prior work [43]. The model uses a convolutional neural network (CNN), more precisely AlexNet [32], as the image encoder and a Multimodal Log-Bilinear language model [44] which is a feed-forward neural network of a single hidden layer as the decoder. This decoder generates the next word based on the linear combination of previous word vectors and encoded image features as context. Meanwhile, the task of machine translation has gained traction with recurrent-neural-network (RNN) based models making their mark [3].

Inspired by the developments in the machine translation domain, Kiros et al [29], further proposed an encoder-decoder architecture based on RNN to leverage the high-dimensional distributed representation space. The model comprises an encoder of a CNN-LSTM setup in which the CNN extracts image features while the long short-term memory network (LSTM) [19], encodes the textual input. These representations are then projected into a multimodal space to achieve a joint embedding. Subsequently, a neural language model, another LSTM network, reads the content vector (image or text embedding) and structure vector (part-of-speech tags) and generates each word conditioned on the auxiliary content and provided structure. The work extensively reports on the multimodal vector space emphasizing that similar underlying concepts should have similar spatial representations. Various advances have been made based on this idea.

Vinyals et al's model [61] is one such advancement that introduced an architecture employed by most of the work that followed. It is explained in Section 2.1.1. One incremental enhancement based on the work by Vinyals et al [61] was that of Xu et al [64]. It applies an attention mechanism to compensate for the dominance of any particular modality. These works also mainstreamed the adaptation of natural language measures like Bilingual Evaluation Understudy (BLEU) [45] and Metric for Evaluation of Translation with Explicit Ordering (METEOR) [34] for evaluating the automatically generated captions. Some of these measures will be detailed under Section 2.4.

2.1.1 Baseline architecture

From the above-mentioned developments, it is evident that a standard underlying architecture has shaped the approach towards image captioning in recent times. Hence, on a conceptual level, it would be fitting to state that all neural image captioning models follow a *de facto* encoder-decoder style architecture [8], as depicted in Figure 2.2. This architectural design is inspired from its earlier success in solving the machine translation task of translating text in one language to another. Neural sequence to sequence models is another consequential term to these models. Concerning the perspective of captioning, the modules of encoder and decoder handle dependent but different objectives. This section details the baseline encoder-decoder setup describing individual components and, thereby, the overall pipeline.

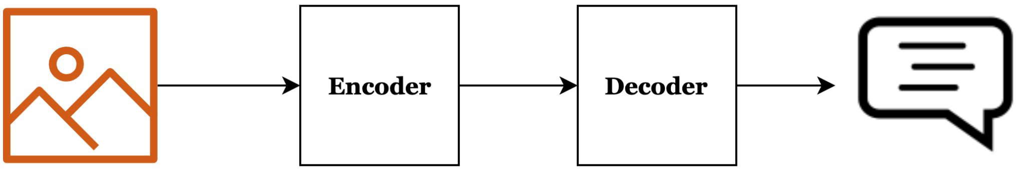


Figure 2.2: Illustration of encoder-decoder framework with visual input to the encoder and textual output from the decoder.

The **encoder**, which is the first component in the play, is typically a CNN owing to their viability in successful detection and summarization of visual semantics. The motivation behind employing an encoder is for the extraction of image features. Image features are usually vector form representations of the actual image, that are necessary for the computational interpretation of the network. The CNN would normally be a pre-trained image classifier network trained with a classification task as the objective, using datasets such as ImageNet [10]. There are a variety of popular well-trained CNNs available, such as AlexNet [32], VGG [56], and Resnet [18]. Each of them serves different purposes, and often Resnet, or deep residual networks, addresses a majority of the use-cases. This popularity of Resnet is due to the network's efficiency in handling the vanishing gradient problem by using residual connections to previous layers as shown in Figure 2.3.

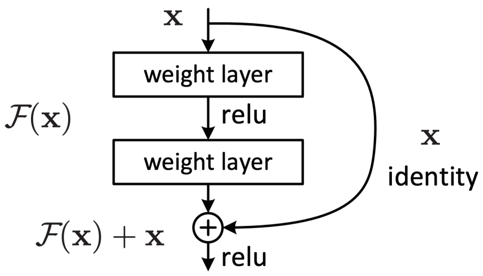


Figure 2.3: Residual learning, a building block; X being the input to a network layer and $\mathcal{F}(X)$, the output. Proposed by [18] for the Resnet architecture.

Another famous image feature extractor is the Inception model [59]. The significant difference with other CNN models is that multiple filters are used to convolve the same input at every level of the network. These different filter sizes make the inception model generalize well to objects of various sizes and provides a range of focus levels. A comparison between variants of the VGG and Resnet architectures is shown in Figure 2.4. Typically there are several preprocessing steps associated with feature extraction using the encoder:

- Resizing the image to match the input specifications of the extractor network, e.g. 256×256 pixels.
- Cropping the classification layers off of the network model and selecting the desired level for extraction, e.g. the penultimate linear layer of Resnet-152 yielding a 2048 dimensional vector.

The **decoder** module of the model is typically a standard recurrent neural network or one of its variants. RNN being inherently sequential is a natural fit for the generation of language or text. Consequently, the decoder networks are autonomously called language models. At every timestep the decoder receives individual word vector embeddings as input. Along with the words, the decoder gets a subject vector (also called a context vector) which is

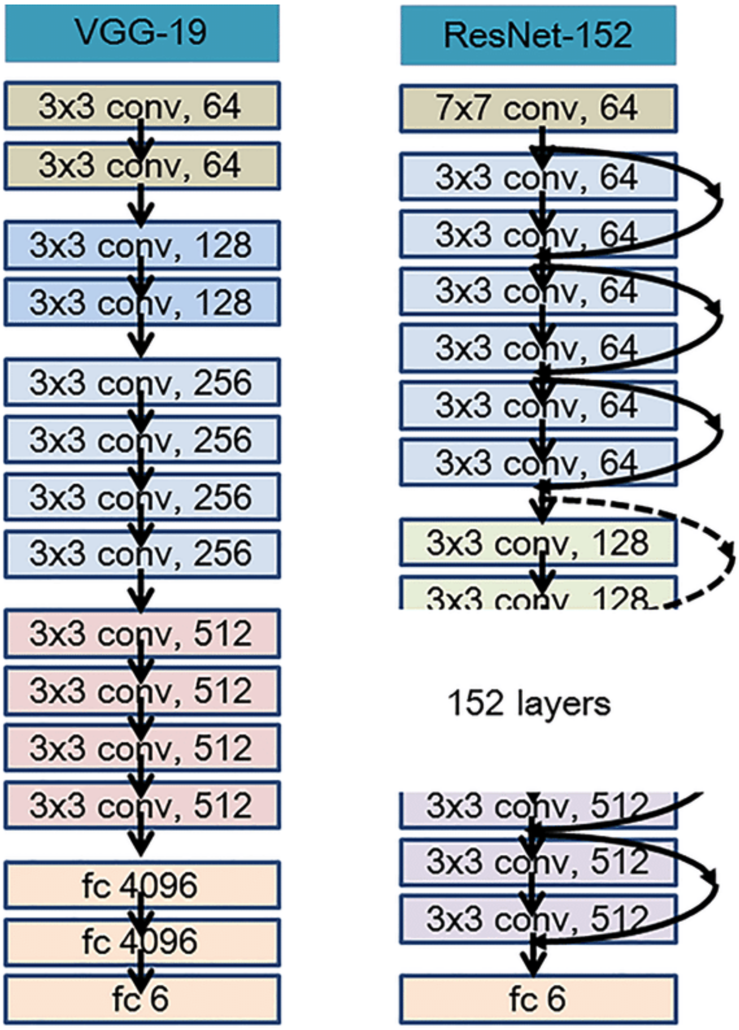


Figure 2.4: Comparison of VGG and Resnet CNN image feature extractor architectures from [53].

essentially the output of the encoder module. The objective of the model is to perceive language conditional on the image subject vector, i.e., to learn a shared space of sentences and images. Words provided as input at each timestep to the network need a representation that can be understood by the model. One approach is the one-hot-vector representation, which is to embed every word as a vector of vocabulary size $\mathbb{R}^{|V|\times 1}$. The one-hot vector structures are very sparse, with 1 at respective word index in the vocabulary and 0 everywhere else. As an improvement, Bengio et al. [6] provided a means of encoding words as distributed continuous representations and termed them word embeddings.

The core idea is to map vocabulary-sized vectors of words to a much smaller-dimensional encoding. Many pre-trained models like Word2Vec [41] and Glove [48] trained on abundant text corpora such as the English Gigaword corpus [16] are readily available. These models provide a projection in which each word maps to a neighborhood of similar words. Additionally, these word-embedder models can be fine-tuned, i.e., trained with the overall language model for vocabulary dependent embeddings. Eventually, the recurrent memory units of the decoder learn the shared representations. This is presented in Figure 2.5. The architecture proposed by Vinyals et al [61] uses LSTM [19] cells to handle the problem of vanishing gradients in traditional RNN blocks. The output from the LSTM units is a vector of arbitrary numbers, fed through a SoftMax layer for obtaining a probability distribution over the vocabulary of words.

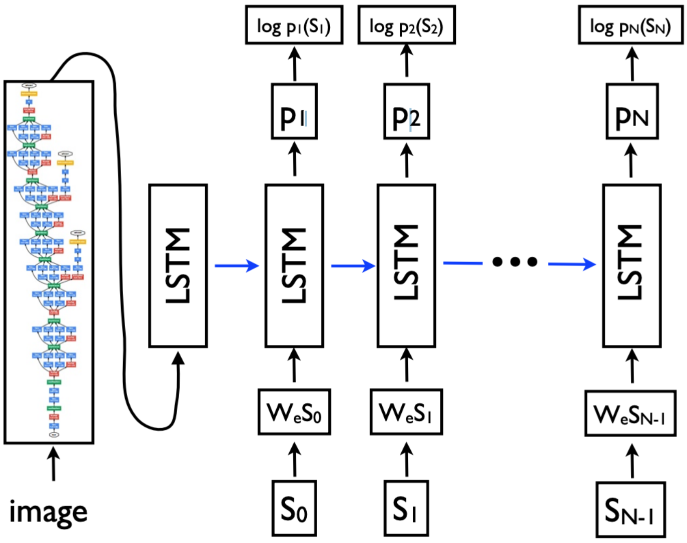


Figure 2.5: Model proposed by Vinyals et al [61].

2.1.2 Dense captioning

Images can be inherently considered as visual scenes, comprising of various elements. Some elements are salient, while others might be less critical existing as background objects supporting and supplying the illustration we perceive. We described in earlier sections the evolution of visual captioning as a task. Classification is what it all started with, wherein there exists a visual input and one corresponding word would be its expected label. Success in the classification task inspired object detection domain of labeling various objects of the image using appropriate words. Gradually image captioning evolved as a task which is about describing an image using complex labels, sentences. Therefore in the space of label complexity and label density as axis, there was scope for a domain that deals with labeling, which is complex yet dense. It is where dense captioning comes in.

Detecting image regions and describing them in a natural language is called dense captioning. The initial work on dense captioning combines the state-of-the-art architectures from both image captioning and object detection domains [25]. Due to the lack of a dedicated dataset for dense captioning, the proposal deals with a two-step procedure. Extracting region-level annotations using a region convolutional neural network (RCNN) [14] and employing multimodal RNN for generating descriptions per region. In essence, the model performs image captioning on cropped areas of the visual input. Flickr8K, Flickr30K [20] and MS COCO [37] are datasets used for reporting results. There are some obvious limitations of this design, from predictions not incorporating the overall context and computational inefficiency to the procedure not following an independent end-to-end style.

Addressing problems from the prior work, the authors of [24] propose a revamped architecture based on the idea of localization as shown in Figure 2.6.

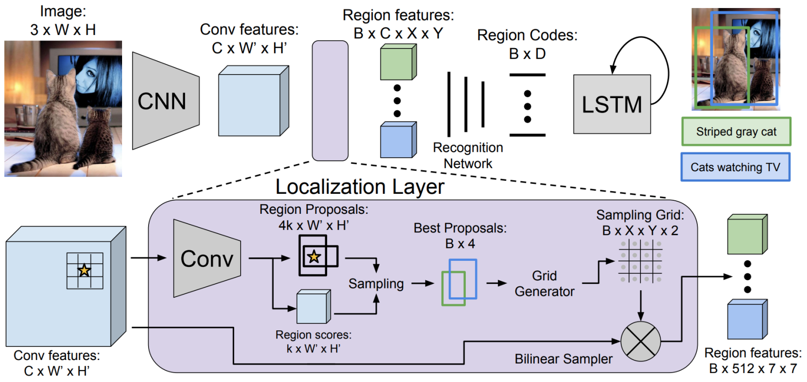
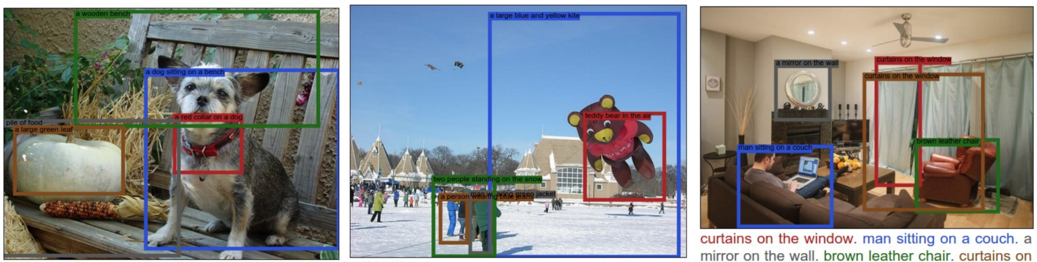


Figure 2.6: The dense captioning architecture proposed by [24].

Heavily inspired by Faster RCNN [50], the fully-convolutional localization layer accepts encoded convolution features from the CNN as input and outputs region co-ordinates and features along with respective confidence scores. A fully-connected recognition network flattens these obtained bounded boxes for further refining the confidence of the proposed regions. Eventually, a standard RNN language model conditioned on the proposed bounding box information generates respective sentences. Taking into account that dense captioning is an open-ended task, the Visual Genome dataset [31] obtained through crowd-sourcing is used. It comprises 94,000 images and 4 million region grounded captions. The work of [24] also describes the feasibility of reversing the task to retrieve related images using region-specific captions. Examples of the generated captions are shown in Figure 2.7.



a red collar on a dog. a dog sitting on a bench. a pile of food. a wooden bench. a large green leaf.

teddy bear in the air. a large blue and yellow kite. man wearing green jacket. two people standing on the snow. a person wearing blue jeans.

curtains on the window. man sitting on a couch. a mirror on the wall. brown leather chair. curtains on the window.

Figure 2.7: Dense captions generated by [24].

2.1.3 Paragraph captioning

Paragraph captioning is an extended use-case of image captioning. In this context, a paragraph is a description comprising more than one sentence. The exact motivation behind conception of paragraph-level descriptions for images could be affiliated with an inherent shortcoming of image captioning. Single sentence captions generated for images predominantly focus on capturing the high-level gist of the visual. Often these single sentence descriptions serve certain use-cases very well while falling short at addressing others. Automatic video subtitling and blind navigation are some of such use-cases in which a mere one sentence description might undermine the objective [36].

Though dense captioning methods explained in the previous section solved this problem to an extent by generating sentences for multiple regions of the visual, they still do not form a coherent whole. Therefore the objective is to produce descriptions with enormous amounts of details. The initial work on paragraph captioning meets this objective and proposes a model by making

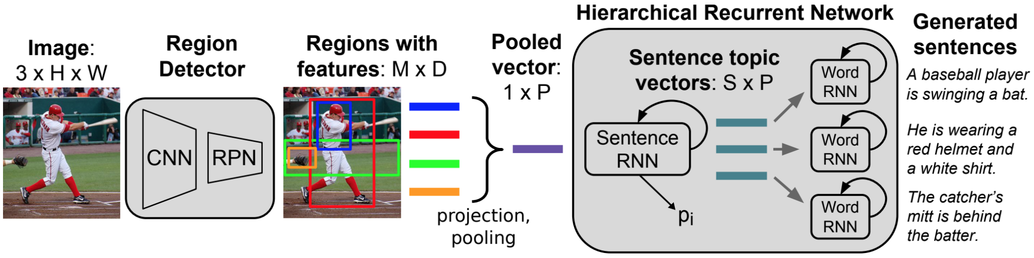


Figure 2.8: Paragraph captioning architecture from [30].

use of the ideas from both image and dense captioning realms [30]. The design relies on the fact that images comprise objects which are granular semantic units to study. It, therefore, employs a region proposal network for detecting regions and eventually pools them into a single image feature vector. Paragraphs comprise sentences that are modular and coherent. Based on that fact, the work in [30] introduces Sentence RNN. This module takes as input the image feature vector and outputs a fixed set of topic vectors along with a halting probability value (halt generation if > 0.5). The topic vectors are inputs to the standard Word RNN modules which generate descriptions.

The Sentence RNN essentially controls the number of decomposable sentences and stopping criteria. The overall model is end-to-end differentiable with CNN and Word RNN modules pre-trained towards the dense captioning task as shown in Figure 2.8. The work also introduces a dataset for the domain, which is a subset of the MS COCO and visual genome datasets. The results outperform all of the baseline models both in terms of the automatic metrics (METEOR, CIDEr) and human evaluation as shown in Figure 2.9. A particular trait of this domain is that the descriptions directly map only to the proposed regions of the image, which allows for interpretability.

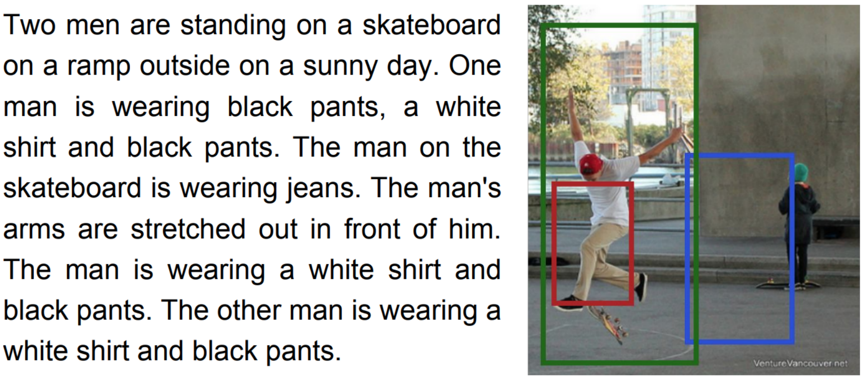

Two men are standing on a skateboard on a ramp outside on a sunny day. One man is wearing black pants, a white shirt and black pants. The man on the skateboard is wearing jeans. The man's arms are stretched out in front of him. The man is wearing a white shirt and black pants. The other man is wearing a white shirt and black pants.

Figure 2.9: A paragraph generated by the model in [30].

2.2 Sequence captioning

Sequence captioning is a colloquial term used in the community for referring to automated captioning tasks dealing with more than one visual. Of course, videos are the most popular sequential data in vision, but tasks involving any form of an ordered collection of images or frames can be called a sequence, e.g.an album of photographs. In the world where photo collages, GIFs, and videos dominate a majority of today's internet traffic, the motivation toward automatically generating descriptions for them is on the surge. Along with all the application domains where automated image captioning shines, sequence captioning opens up gates to additional areas. With the potential for being able to comprehend events in real-time, sequence captioning can enable human-like interaction with service robots. Smart assistants can perceive environments and converse back in more coherent and meaningful ways. There are, however, numerous challenges affiliated with sequence captioning to overcome before achieving such advancements.

Videos or sequences of images are, in essence, elaborated visual information captured for convenience. Therefore there is scope for a lot of redundancy, unnecessary details, and misleading contexts. These inherent properties bring in difficulties related to understanding subjects, tracking various objects, capturing the causality of multiple events, speed, direction, and overlap of the plausible scenes. Classical methods utilized the idea of selecting salient elements from the data and thereby applying rule-based templates for learning and generating descriptions. Indeed these methods quickly proved inadequate as the datasets evolved and grew in diversity. With the success of deep learning, researchers tried the divide-and-conquer strategy of projecting this task into the image captioning realm and failed, owing to the complications and inherent differences above mentioned.

Valid object recognition is obviously at the core of the task, but aspects related to activity detection such as event inference, salient relationships, and adequately addressing diversity, influence the quality of a model. Nevertheless, there has been significant research interest that has continuously been growing towards sequence captioning and related sub-fields such as video captioning, Section 2.2.1, visual storytelling, Section 2.2.2, video question-answering, and dense video descriptions. Recent directions toward solving these hurdles such as audio modality accompanying videos and incorporating prior external knowledge are fascinating.

2.2.1 Video captioning

Video captioning is traditionally a task of conveying information about a video clip as a whole by automatically generating a single natural sentence [1]. Over the years, the goal of video captioning changed to comprehending spatio-temporal information in a video clip as language. Related sibling fields include video descriptions, paragraph generation for clips, and video question-answering. As mentioned, video captioning was initially approached using the method of detecting subject, verb, and object (SVO) from the visual, and combining them using sentence templates [4]. Nonetheless, with the advancements in deep learning and progress within the image captioning domain, several architectures have been proposed to solve the problem of captioning videos. A basic underlying framework is summarized in Figure 2.10.

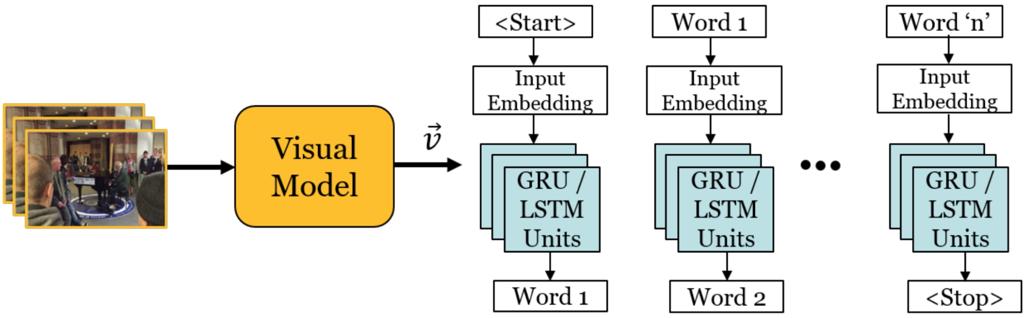


Figure 2.10: A baseline framework for video captioning from [1].

The visual model is usually a two or three-dimensional CNN depending on the nature of the visual data. A three dimensional CNN serves well when there is scope for change in geometry of objects across frames, or there are action-oriented details like hand gestures. The computational cost of training the model increases when a three dimensional CNN is employed. Another factor that could influence the choice of the visual encoder model is the dataset on which it was pre-trained. It might not be optimal to choose a 3D CNN trained on a dataset of wildlife clips when the data of the task at hand is about indoor environments. Hence, 2D CNN is used for many use-cases, and mean or maximum pooling techniques are applied for obtaining a single vector. To address long video clips, sampling key frames has been shown to perform well [55].

The encoded visual features are passed on to a standard RNN language model for the generation of a sentence. Some methods generate multiple descriptions for a video clip based on the pre-analysis of the ground truth [52].

Describing long video clips remains a hard problem even today due to the lack of a dataset with adequate and diverse vocabulary, which is a prerequisite for detecting actions. Most of the proposed encoder-decoder models only differ in terms of the type of video encoder and minor variations in initializing the language model using the visual features. Some later methods introduced attention mechanisms while maintaining the same underlying design [65]. In recent times, the actual evolution has happened in terms of the task itself. Dense video captioning has come into focus with the availability of new datasets. Several possible scenarios of captioning video frames are shown in Figure 2.11.

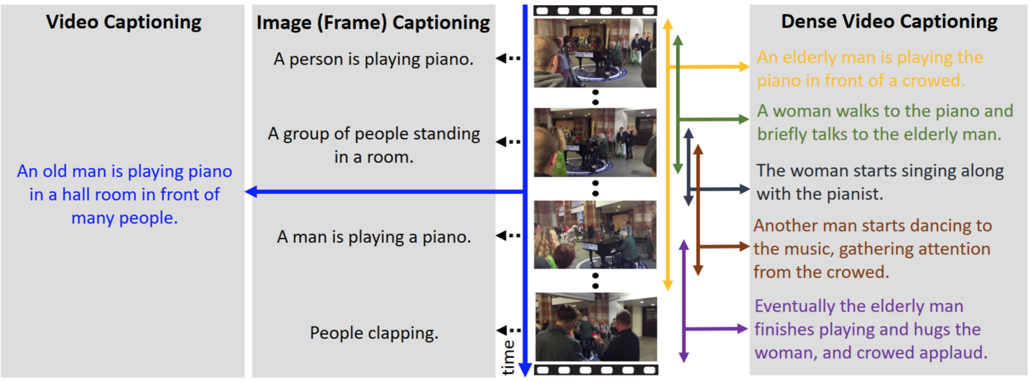
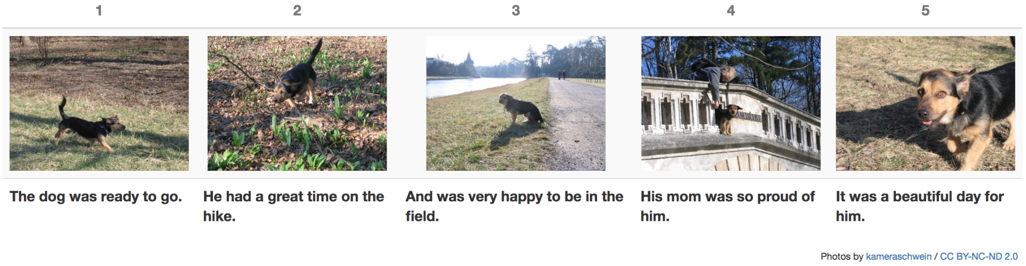

Video Captioning
An old man is playing piano in a hall room in front of many people.
Image (Frame) Captioning
A person is playing piano.
A group of people standing in a room.
A man is playing a piano.
People clapping.
time
Dense Video Captioning
An elderly man is playing the piano in front of a crowed.
A woman walks to the piano and briefly talks to the elderly man.
The woman starts singing along with the pianist.
Another man starts dancing to the music, gathering attention from the crowed.
Eventually the elderly man finishes playing and hugs the woman, and crowed applaud.

Figure 2.11: Illustration of differences within the video captioning realm [1].

2.2.2 Visual storytelling

All the sections above outline the time-frame of evolution, respective motivations, and current status of captioning as a task. For the most part, it has been some inadequacy that drove the need and created space for a new domain. It is no different in the case of visual storytelling. Sequence descriptions, dense or sparse, lengthy or concise, are principally restrained under the cover of naïveness. The objective of a well-trained video captioning model is to perceive best the visual and merely produce language comprising relevant objects and attributes. These models lack imaginative power and inferential intelligence that can drastically enhance vision-related use-cases driven by automatically generated language. The necessity to generate narrative style texts for image sequences that reflect experiences, rather than simple elements, has motivated the task of visual storytelling. Therefore the problem statement of visual storytelling is, given a sequence of images (or frames of a video) as input, learning a model, that can output a story with abstract, subjective aspects while contextualizing the input sequence. Visuals in the input sequence typically adhere to an ascending time-frame order. Figure 2.12 exemplifies a data sample from the visual storytelling dataset (VIST) [22] which will be detailed under Section 2.3.



The dog was ready to go.

He had a great time on the hike.

And was very happy to be in the field.

His mom was so proud of him.

It was a beautiful day for him.

Photos by kameraschwein / CC BY-NC-ND 2.0

Figure 2.12: An example sequence from VIST and a respective generated story [22]. Words like ***great***, ***proud***, ***ready*** can be tagged as subjective concepts within the narrative.

The initial work on visual storytelling used NYC and Disney image sets crawled from blog posts over the web [46]. There were multiple bottlenecks to address to use these datasets for the storytelling task. Lack of annotations and non-event nature of the extracted images rendered the datasets futile. Huang et al [22] released the VIST dataset, which will be described under Section 2.3. The dataset was created exclusively for the visual storytelling task. Along with the dataset, they published a baseline model, see Section 3.1, and reported evaluation scores of the stories generated by their model. The baseline architecture followed the encoder-decoder structure, extending the Show and Tell [61] image captioning model. The VIST dataset was made public as a part of the storytelling competition which led to the conception of several other approaches towards the idea of visual storytelling. Gonzalez-Rico et al [15], as will be elaborated in Section 3.3, and Smilevski et al [57], as will be elaborated in Section 3.2, were some of the first methods to be proposed for visual storytelling. Both approaches extend the baseline architecture, better fitting the VIST dataset.

2.3 VIST dataset

The VIST dataset includes 10,117 Flickr albums with 210,819 unique photos. The release comprises three tiers of language for the same set of images; **descriptions of images-in-isolation** (DII), **descriptions of images-in-sequence** (DIS), and **stories for images-in-sequence** (SIS). Shortlisted albums were from Flickr, which had storyable events, like *John's birthday party*, or *Friends' visit*. Subsequently, crowd-workers through Amazon Mechanical Turk extracted stories for grouped photo-sequences within the albums as depicted Figure 2.13.

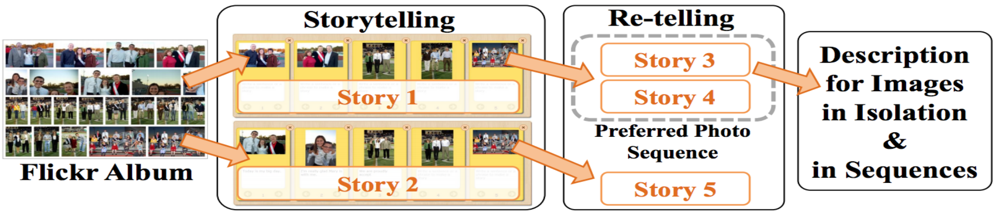


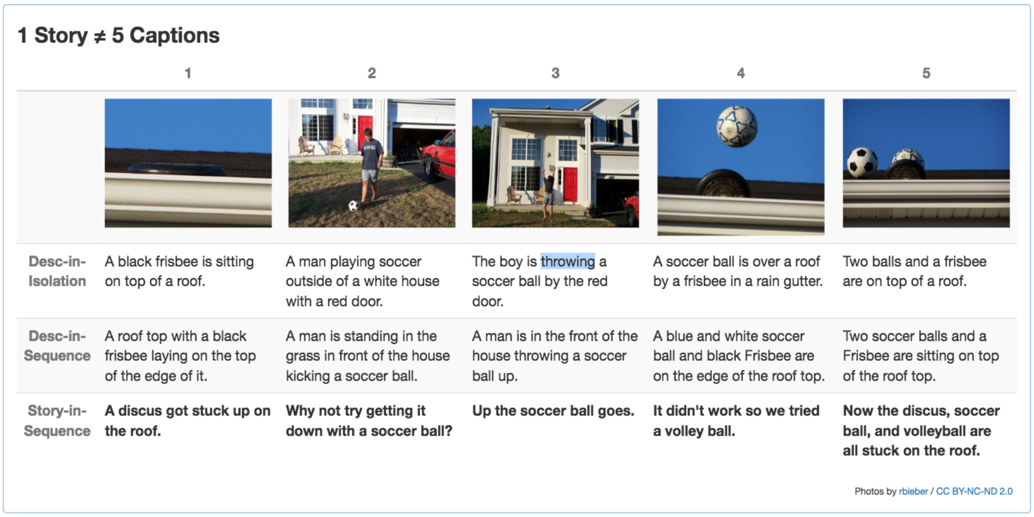
Figure 2.13: Dataset crowd-sourcing workflow of the VIST dataset from [22]. For each album two workers perform storytelling and three workers perform retelling on the photo-sequences selected in the storytelling phase.

The obtained stories were post-processed by tokenizing them using the CoreNLP toolkit [38] to replace people's names, specific locations, and other identifiers with generic de-identified tokens. Eventually, the final data release comprised training, validation, and test splits following 80%, 10%, 10% proportions, respectively. The DIS data tier uses the same procedures and interfaces with an additional instruction for the workers to follow MS COCO [37] description styles, like *"describe all the essential parts"*. The DII data tier leverages the complete MS COCO captioning interface.

In the SIS data tier, each sequence has five images with corresponding descriptions, which together make up for a whole story. Furthermore, for each Flickr album, there are five permutations of a selected set of its images. In the overall available data, there are 40,071 training, 4,988 validation, and 5,050 usable testing stories. Figure 2.14 visualizes a data sample. The VIST dataset is not perfect. It has some inherent flaws like character bias, baseless abstract words, and limited size vocabulary. Nevertheless, being the only available straightforward dataset for the storytelling task, it is understandable that many published models purvey to better-fitting the VIST data.

2.4 Evaluation measures

Human evaluation of automatically generated or translated text is the gold standard for judging the robustness of a model. However, it is impractical, primarily owing to the expensiveness of human labor. Additionally, human judgment is not reusable or generalizable to minor perturbations in use-cases.



1 Story ≠ 5 Captions

	1	2	3	4	5
Desc-in-Isolation	A black frisbee is sitting on top of a roof.	A man playing soccer outside of a white house with a red door.	The boy is throwing a soccer ball by the red door.	A soccer ball is over a roof by a frisbee in a rain gutter.	Two balls and a frisbee are on top of a roof.
Desc-in-Sequence	A roof top with a black frisbee laying on the top of the edge of it.	A man is standing in the grass in front of the house kicking a soccer ball.	A man is in the front of the house throwing a soccer ball up.	A blue and white soccer ball and black Frisbee are on the edge of the roof top.	Two soccer balls and a Frisbee are sitting on top of the roof top.
Story-in-Sequence	**A discus got stuck up on the roof.**	**Why not try getting it down with a soccer ball?**	**Up the soccer ball goes.**	**It didn't work so we tried a volley ball.**	**Now the discus, soccer ball, and volleyball are all stuck on the roof.**

Photos by rbieber / CC BY-NC-ND 2.0

Figure 2.14: Sample descriptions of the DII, DIS and SIS language tiers of the VIST dataset from [22].

To handle this bottleneck, Papineni et al [45] proposed the BLEU metric focused on evaluating machine translated hypotheses. The rationale behind is to mimic the human way of judging the relevance of a sentence, given the expected true sentence. Considering the candidate translation length c and reference corpus length r, BLEU score is computed as follows:

$$\text{BLEU-}N = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \quad \text{where} \tag{2.1}$$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} .$$

N is the range of n-grams with p_n denoting precision (overlap between reference and candidate sentences) and w_n representing the customizable importance weights summing to 1. The motivation behind the brevity penalty BP is to account for length matching between the references and hypothesis. BLEU metric ranges from 0 to 1, and for this thesis, the reported scores are BLEU-4 symbolizing the consideration of n-grams from 1 to 4 during scoring.

Nevertheless, certain shortcomings of the BLEU metric surfaced overtime. Importantly, BLEU does not account for recall, which might result in misleading scores. To address the challenges with BLEU, METEOR [34] automatic evaluation metric was conceived. Alignments between words and phrases in METEOR are based on the stem, synonym, or paraphrase matching between hypothesis-reference pairs. The exact computation process is summarized as follows:

$$\begin{aligned}\text{Precision} &= \frac{\text{number of matching unigrams}}{\text{number of unigrams in predicted sequence}} \ , \\ \text{Recall} &= \frac{\text{number of matching unigrams}}{\text{number of unigrams in reference sequence}} \ , \\ F_{mean} &= \frac{10 \ \cdot \ \text{Precision} \ \cdot \ \text{Recall}}{\text{Recall} + 9 \ \cdot \ \text{Precision} \ \cdot \ \text{Recall}} \ , \\ \text{Fragmentation penalty} &= 0.5 \ \cdot \ (\frac{\text{number of matching chunks}}{\text{number of matching unigrams}})^3 \ , \end{aligned} \tag{2.2}$$

$$\text{METEOR} = F_{mean} \ \cdot \ (1 - \text{Fragmentation penalty}) \quad .$$

F_{mean} weights recall nine times more than precision and combines them using the harmonic mean. Fragmentation penalty accounts for the correlation of longer matches by considering the ratio of contiguous chunks to unigrams. Additionally, the metric provides parameters for handling punctuation, tokenizing and weighting modules. These options are customizable to reflect several human biases for tasks such as machine translation and captioning. At sentence level, it is empirically debated across the community, that assessment using METEOR is better than previous scorers such as BLEU [45]. Arguably, there are better scorers compared to METEOR, such as CIDEr [60], which considers the vocabulary of the overall corpus during evaluation. However, this thesis work largely utilizes METEOR, adhering to the norm across publications on visual storytelling.

Chapter 3

Comparison of existing models

Since the conception of visual storytelling in 2016, various proposals addressing the task came out. Some of them were part of the 2018 Visual Storytelling Challenge[1] and others are extended approaches based on the challenge results. This chapter details the most popular methods in the order of their publishing time by discussing model architectures, implementations, and respective results. The next chapter discusses remodeled methods, experiments, and respective results. Evaluation scores pertaining all the experiments from the current and the following chapter, are consolidated in Table 4.1 under Section 4.3.

3.1 Visual storytelling baseline

Huang et al [22] mainstreamed the domain of visual storytelling with their work. They released the VIST (visual storytelling) dataset (detailed in Section 2.3) which is the first and only full-fledged dataset available for the visual storytelling task till date. Along with the dataset, this work presented various measures and results of baseline experiments on the task. The primary intent was to introduce the problem statement of visual storytelling rather than solving it. Therefore they proposed a sequence-to-sequence recurrent neural network (RNN) architecture shown in Figure 3.1, extending the single-image captioning technique of [11] and [61] to multiple images.

The encoder module reads the image sequence features extracted using a pretrained convolutional neural network based feature extractor. The images in the sequence were read in the reverse order. The authors do not explicitly provide reasoning behind such reversal, but it can be seen as a way to incul-

[1] http://www.visionandlanguage.net/workshop2018/index.html#challenge

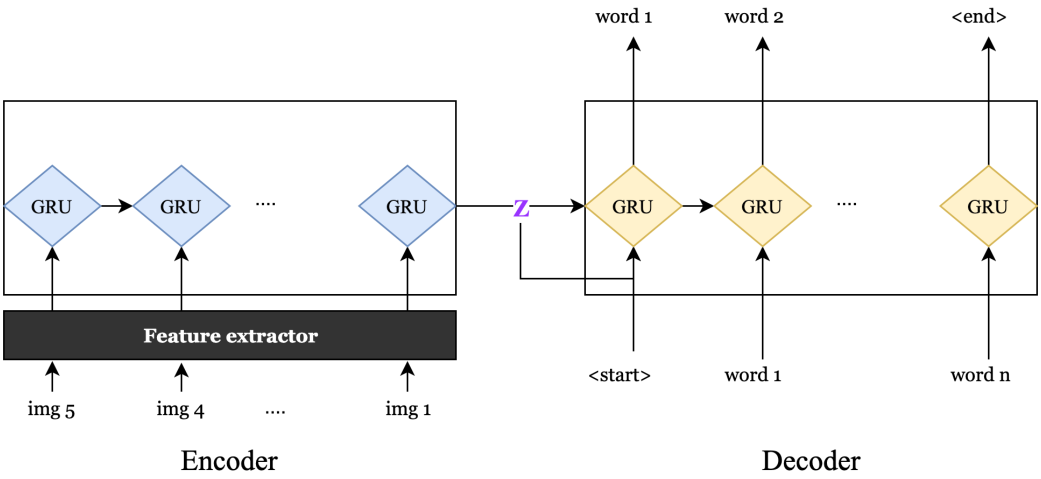


Figure 3.1: Visual storytelling baseline architecture (inferred based on the description in [22]).

cate a futuristic dependency between events of the sequence. The features then sequentially pass through the RNN, yielding a context-vector Z as shown in Figure 3.1. The context vector is then passed both as the initial hidden state and as the first input to the decoder RNN module by concatenating it with the `<start>` token embedding. The decoder module then learns to produce the story word-by-word, at every time-step. Gated Recurrent Units (GRU) [9] were used for both the image-sequence encoder and the story decoder. The publication does not mention other implementation details or configurations related to model training.

Given the complexity of the task, the most reliable means of assessing the generated stories is through human judgement. Nevertheless, for computational efficiency and particularly for standardizing the aspect of benchmarking the authors employ and report several automatic evaluation metrics such as BLEU, METEOR, perplexity and other vocabulary diversity measurements explained under Section 2.4. However, for the purpose of comparison, only METEOR scores are considered in this thesis, owing to the *de facto* standardization by the visual storytelling challenge and community. The best reported METEOR score of the proposed model on the VIST dataset was 0.31. The split of the dataset on which this scoring was performed was, however, unreported.

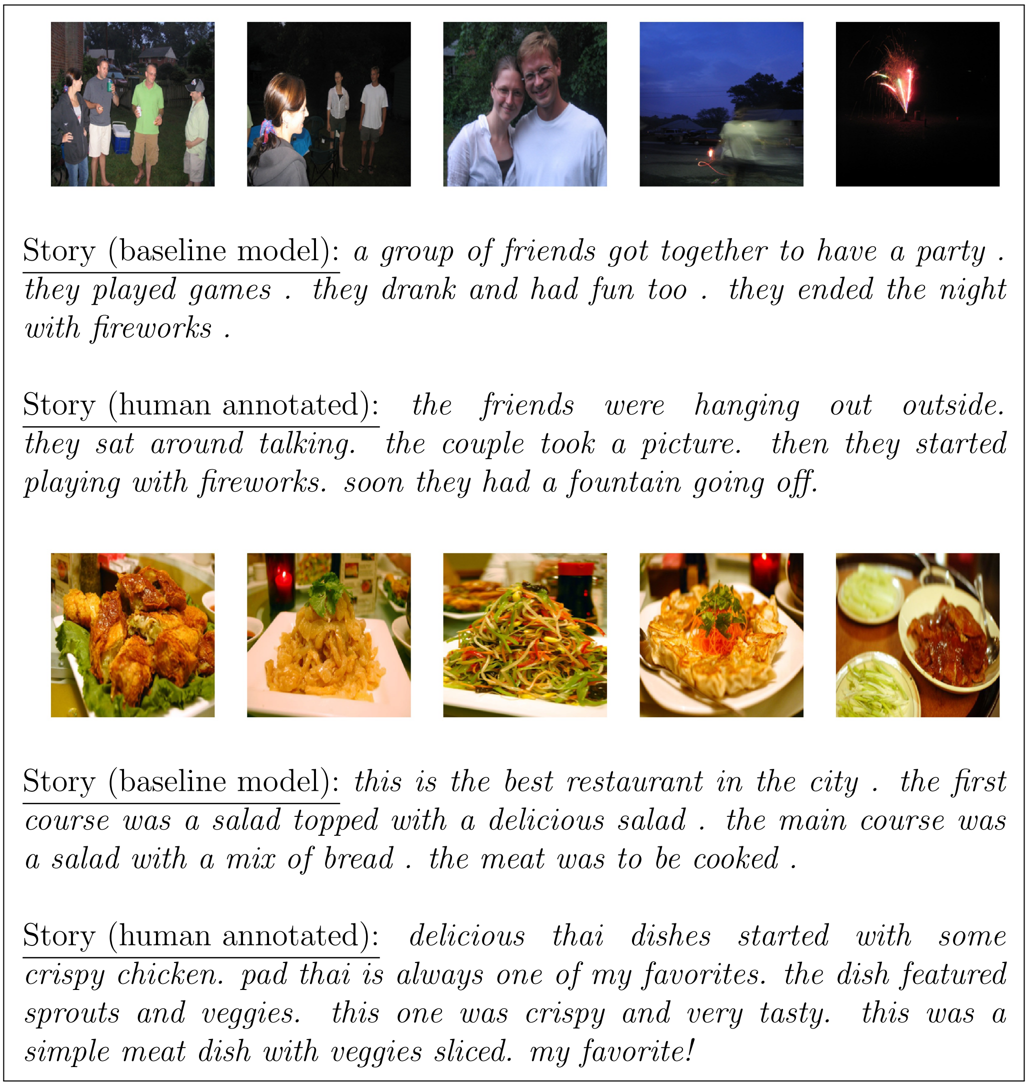
To validate the claims and findings in the paper, the baseline architecture was implemented for this thesis from scratch in PyTorch [47]. Preliminary information about some of the model parameters was available from the FAQ section of the dataset. VGG16 [56] was used as the pretrained image feature extractor (without fine tuning). The encoder and decoder were 1,000 dimensional GRU networks without weight-sharing. Embedding layer was employed and learned for target word embedding, with dimension sizes varying between 256 and 1024. Vocabulary was constrained to consider only words that occur three or more times in the training stories. Other words were mapped to the `<unk>` token. Dropout of 0.5 was used between all the intermediate layers of the model. Learning rate of 0.0001 was incorporated for 80 epochs and validation was performed on the held-out dataset after every epoch. Training and validation were performed in batches of size 64. To account for the variation in story lengths within each batch, zero-padding was used. Eventually, using greedy sampling based inference, our implementation obtained a METEOR score of 0.216 on the entire validation split compared to the 0.31 in [22]. Sample results can be seen in Figure 3.2.

3.2 Stories for Images-in-Sequence by using Visual and Narrative Components

From Section 3.1 it is evident that the baseline model aligns an image sequence with the entirety of its respective story. Conversely, Smilevski et al [57] present a multi encoder approach. The architecture is displayed in Figure 3.3. The primary intention is to align every sentence in the story with incremental subsets of image groups from the respective image sequence. This means that a sentence-story is generated per image while considering a group of appropriate number of images from the same sequence. The image sequence encoder RNN models the image feature vectors of the sequence making it similar to the encoder module described in Section 3.1. The previous sentence-story encoder RNN learns the temporal behavior of words in the sentence-story generated for the previous image of the sequence.

Both encoders generate fixed-length vector representations, one for the image-sequence and one for the previous sentence-story. The two representations are then concatenated together to form a joint embedding, which is used as an initial hidden state for the decoder module. The authors argue that such a setting of encoders would allow the image sequence to have more impact on the generated text. However, the decoder RNN module behaves just like the baseline model decoder, detailed in Section 3.1.

The code base of the multi encoder model is publicly available. We replicated the experimental runs to compare and analyze the reported results. The implementation uses fc7 vectors from AlexNet [32] feature extractor network to describe the images. The created vocabulary comprised the most



Story (baseline model): *a group of friends got together to have a party . they played games . they drank and had fun too . they ended the night with fireworks .*

Story (human annotated): *the friends were hanging out outside. they sat around talking. the couple took a picture. then they started playing with fireworks. soon they had a fountain going off.*

Story (baseline model): *this is the best restaurant in the city . the first course was a salad topped with a delicious salad . the main course was a salad with a mix of bread . the meat was to be cooked .*

Story (human annotated): *delicious thai dishes started with some crispy chicken. pad thai is always one of my favorites. the dish featured sprouts and veggies. this one was crispy and very tasty. this was a simple meat dish with veggies sliced. my favorite!*

Figure 3.2: Image sequences and corresponding stories generated by the Visual storytelling baseline [22].

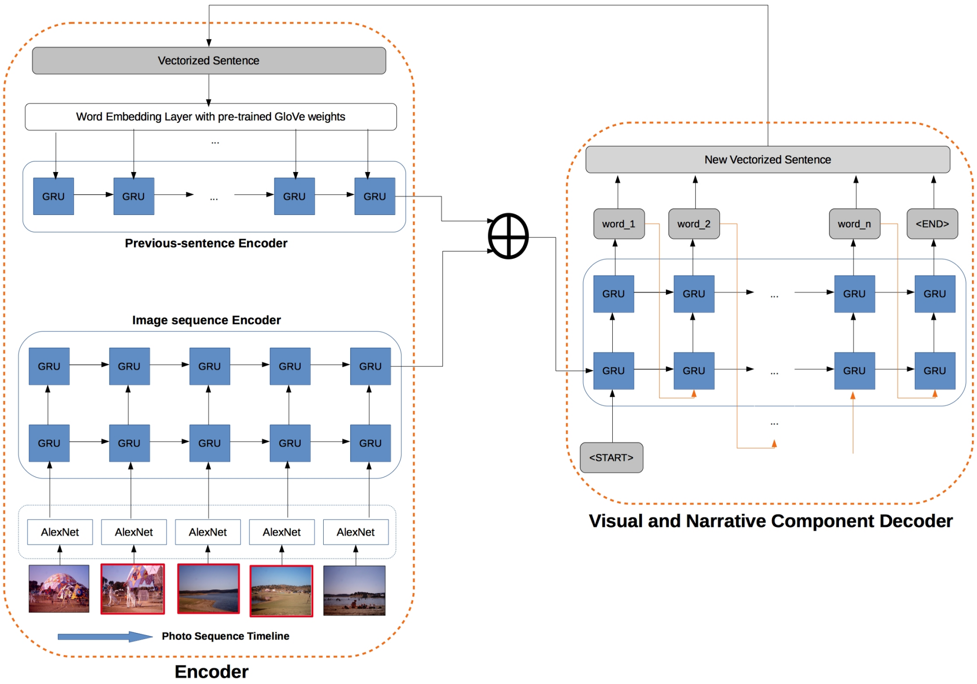


Figure 3.3: Multi encoder model architecture from [57].

frequent words, appearing four or more times. Based on the analysis of training data stories, the maximum length of sentences was chosen as 20. Before the sentence vectors entered the previous sentence-story encoder and the current sentence-story decoder, they were passed through an embedding layer. The embedding layer uses GloVe [48] pretrained word vectors. This transforms the sentences from 22 word vectors, two words for the `<start>` and `<end>` tokens, to a vector of 22 word embedding vector of vectors by mapping the input words to corresponding pre-learned representations.

The image-sequence encoder RNN comprised 1,024 neurons and the previous-sentence encoder RNN 512 neurons. As the outputs from both the encoders are concatenated to represent the context, the decoder is a 1,536 dimensional RNN. Categorical cross entropy was used as a loss function. The learning rate was set to 0.0001 throughout the training. Adam optimization algorithm was used during back-propagation. To avoid over-fitting the network to the data, during the training process dropout of 0.3 on the input layer and 0.5 on the pre-output layer was applied to regularize. The number of previous images to consider was an additional parameter to choose for the image-sequence encoder. Empirically, the model performed best when the number of previous images parameter was set to three. A METEOR score of 0.225 was obtained on the validation split of the dataset. Sample results can be seen in Figure 3.4.

For the purpose of comparison, same image sequences from results of Section 3.1 were utilized. The blue colored words indicate good associativity level of the model with regard to emphasizing visual data while the red colored words suggest otherwise. From a storytelling standpoint, these associativity levels can be seen as trade-off between commonsense and creativity. This model is interesting as it was the first model for visual storytelling to focus on learning relationships between visual and textual data. The authors argued that the automatic evaluation metrics can merely differentiate only between a good and an obviously bad generated story and resorted to human evaluation.

3.3 Contextualize, Show and Tell: A Neural Visual Storyteller

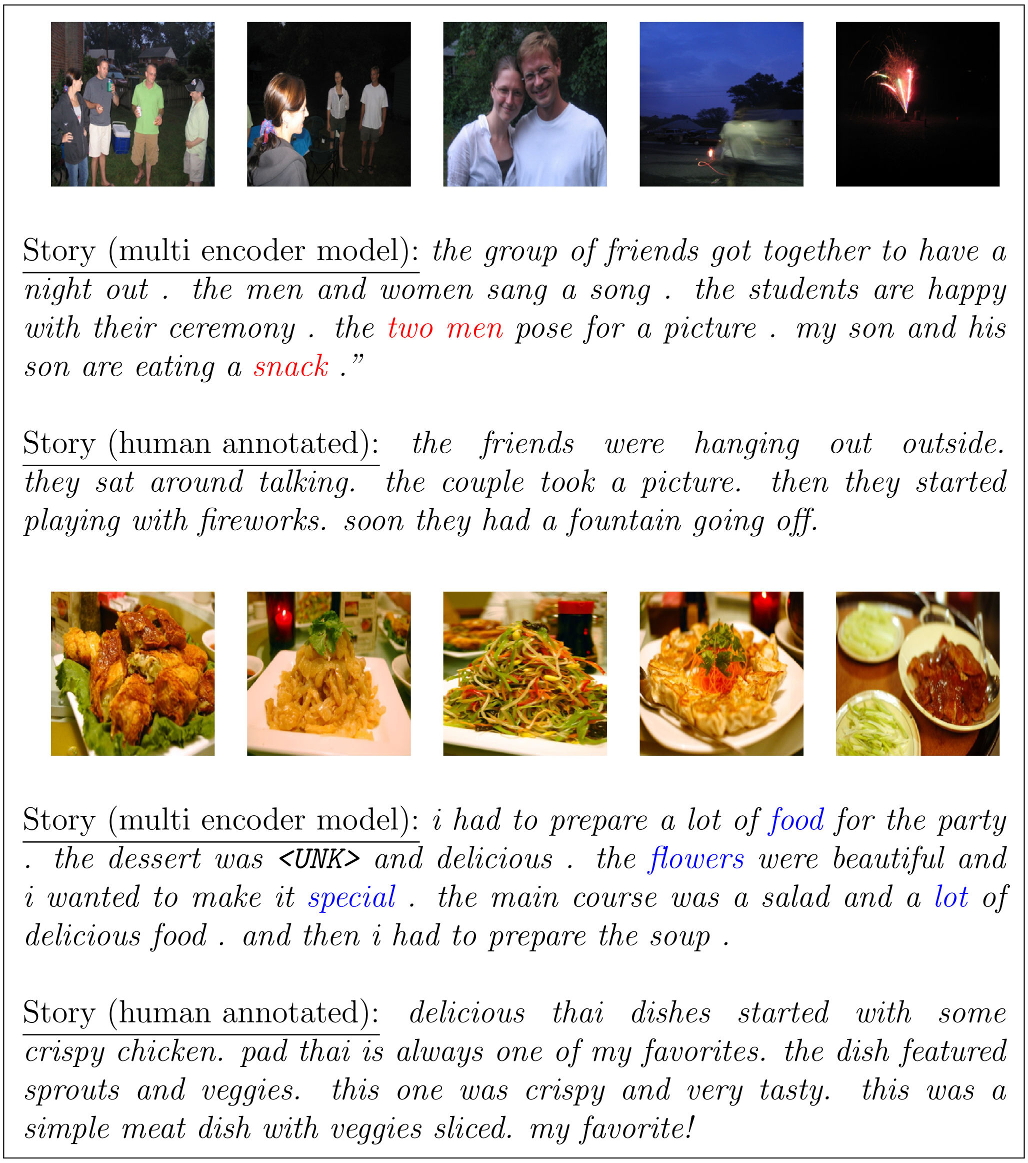
In 2018, NAACL organized a challenge on visual storytelling[2]. The objective for the participants was to create AI models that can generate stories, sharing human experience and understanding. The internal track of the challenge uses only the VIST dataset as mentioned in Section 2.3, while the external track allowed for leveraging any publicly available dataset. There were two parts of evaluation, i.e., automatic and human. Crowd-sourced survey methods were employed for human evaluation and METEOR version 1.5 was used for automatic evaluation. The underlying aspects taken into consideration during evaluation include the following:

- **Structure and coherence** of the story dealing with grammatical body and hierarchy of the sentences.

- **Focus** on the sentence comprehensiveness, context preservation and overall appropriateness of details from image sequences (visual modality).

Gonzalez-Rico and Pineda [15] were the winning team of the VIST 2018 challenge. In the internal track of the competition, they had the leading METEOR score and human judgement based on the above aspects. We reference their model as multi-decoder architecture, present it in Figure 3.5 and discuss it in this section.

[2] http://www.visionandlanguage.net/workshop2018/index.html#challenge



Story (multi encoder model): *the group of friends got together to have a night out . the men and women sang a song . the students are happy with their ceremony . the two men pose for a picture . my son and his son are eating a snack ."*

Story (human annotated): *the friends were hanging out outside. they sat around talking. the couple took a picture. then they started playing with fireworks. soon they had a fountain going off.*

Story (multi encoder model): *i had to prepare a lot of food for the party . the dessert was <UNK> and delicious . the flowers were beautiful and i wanted to make it special . the main course was a salad and a lot of delicious food . and then i had to prepare the soup .*

Story (human annotated): *delicious thai dishes started with some crispy chicken. pad thai is always one of my favorites. the dish featured sprouts and veggies. this one was crispy and very tasty. this was a simple meat dish with veggies sliced. my favorite!*

Figure 3.4: Image sequences and corresponding stories generated by the multi encoder model [57].

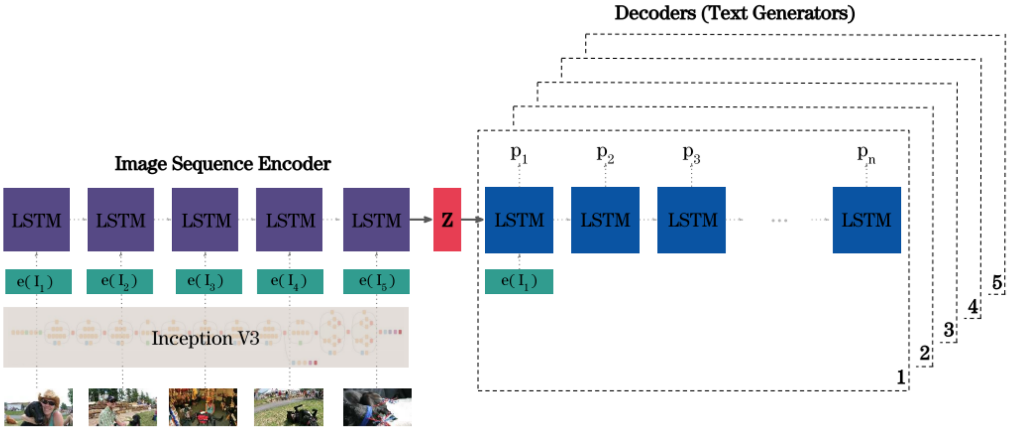
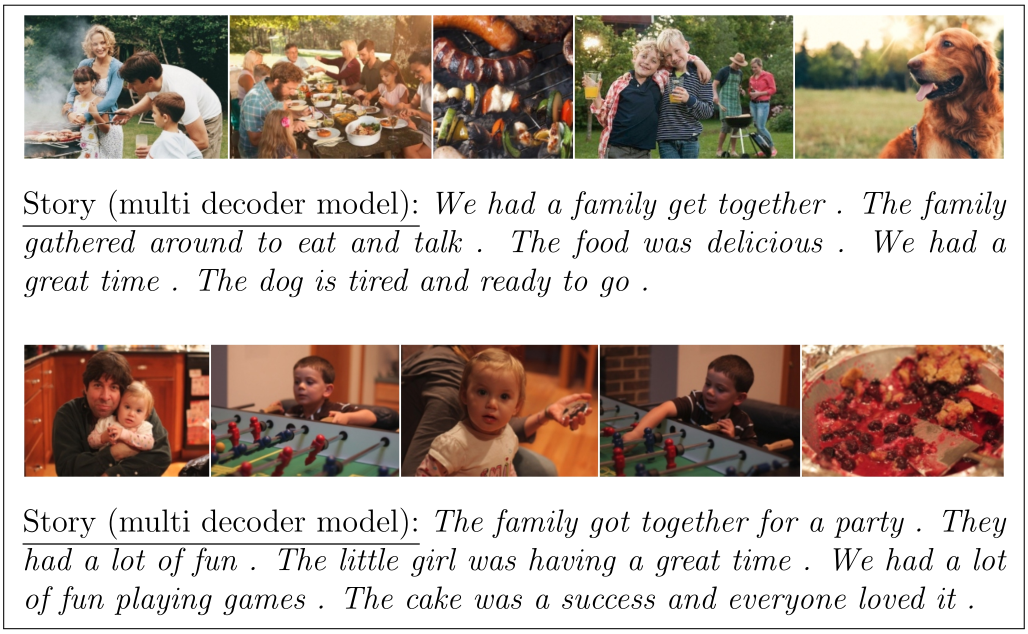


Figure 3.5: Multi decoder model architecture proposed by [15].

The architecture follows the encoder-decoder style extending the model by Vinyals et al., 2015 [61]. The encoder, the initial component of the model is a RNN and specifically an LSTM for summarizing the sequences of images. It is very similar to the encoder module presented in Section 3.1, reading each image from the sequence, as input at every timestep. Eventually, the last hidden state of the encoder represents a contextualization of the images in the sequence. For representing the images, the authors use the Inception V3 [59] feature extractor model without fine-tuning.

The novel aspect of this model are the decoders. Owing to the presence of five images per sequence in the VIST dataset, see Section 2.3, the model comprises five decoders. Each decoder is responsible for generating a sentence-story for the respective image in the sequence and finally the combined sentence-stories make up for a story. These multiple decoders are independent of each other, i.e., they do not share any parameters (weights). As stated above, the last hidden state of the encoder is passed on as the initial hidden state to all the decoders. For each of the decoders, the first input is the respective image feature extracted using a pretrained feature extractor. The authors argue that the motivation behind such a strategy is to provide the decoder with the context of the whole sequence and the content of the image at a particular timestep (i.e. global and local information) to generate the corresponding text that will contribute to the overall story. The individual decoders themselves behave in the same way as the decoder module of the baseline model, Section 3.1. A major distinction is that the initial input to the multi decoders is the respective image content instead of the context of the sequence.

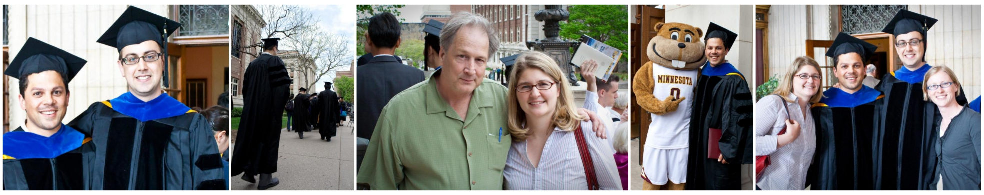
Although some of the implementation details were unavailable from the publication, we implemented the architecture by performing hyper-parameter tuning and tried various ways of back-propagation. Later, the authors open sourced the implementation setup. Word2Vec [41] was used for embedding the text and categorical cross-entropy was used for loss calculation. A learning rate of 0.0005 was used along with stochastic gradient descent optimization, scheduled decay by 0.5 every 8 epochs, and gradient clipping with a threshold of 5. The input and output dimensionality of the LSTM was set to be 512. The overall model was trained for 500000 steps and based on the number of samples from the dataset, which translates to 120 epochs. The model achieves a METEOR score of 0.34. Stories generated by the model for external (non VIST) sequences of images can be seen in Figure 3.6 and for a VIST sequence in Figure 3.7. From the results, it can be observed that the model adheres to the tone and topic of the narrative very consistently.



Story (multi decoder model): *We had a family get together . The family gathered around to eat and talk . The food was delicious . We had a great time . The dog is tired and ready to go .*

Story (multi decoder model): *The family got together for a party . They had a lot of fun . The little girl was having a great time . We had a lot of fun playing games . The cake was a success and everyone loved it .*

Figure 3.6: Image sequences and corresponding stories generated for a non-VIST image sequence by the multi decoder model [15].



Story (multi decoder model): *Today was graduation day . The students were excited . My parents were happy too . He was very happy to be graduating . Everyone was so proud of him .*

Story (human annotated): *Today was graduation, and Schyler was extremely happy. However he was nervous about what the future would bring. His parents assured him that he would do well in life. That helped as little. Of course when Benny the squirrel gave him life advice his whole demeanor turned happily. Schyler is now ready for life, after the first big chapter ending high school.*

Figure 3.7: Image sequences and corresponding stories generated for a VIST image sequence by the multi decoder model [15].

3.4 GLocal Attention Cascading Networks for Multi-image Cued Story Generation

The primary difference between standard captioning and sequence captioning, as explained in Section 2.2, is the aspect of overall context. Sequence captioning makes the problem more challenging and the methods compared in Sections 3.2 and 3.3 try to address it. As part of the VIST 2018 challenge, another idea was the GLAC net [27]. The authors try to address the difficulty of maintaining the specificity of one image while still maintaining the dominance of the overall image sequence context. This was the first architecture to bring in attention mechanism to the visual storytelling realm. However, at the core, it follows the familiar encoder-decoder components which are detailed in the following part of this section. The architecture is shown in Figure 3.8.

Similar to other methods, Resnet-152 [18] was employed for extracting features from the visuals. These features are then sequentially passed through a bi-directional LSTM encoder network. The encoder component functions similar to the one described in Section 3.1, but in both directions of the input sequence. Therefore encoded vectors at each timestep of the encoder comprise context of the past and the future images in the sequence with

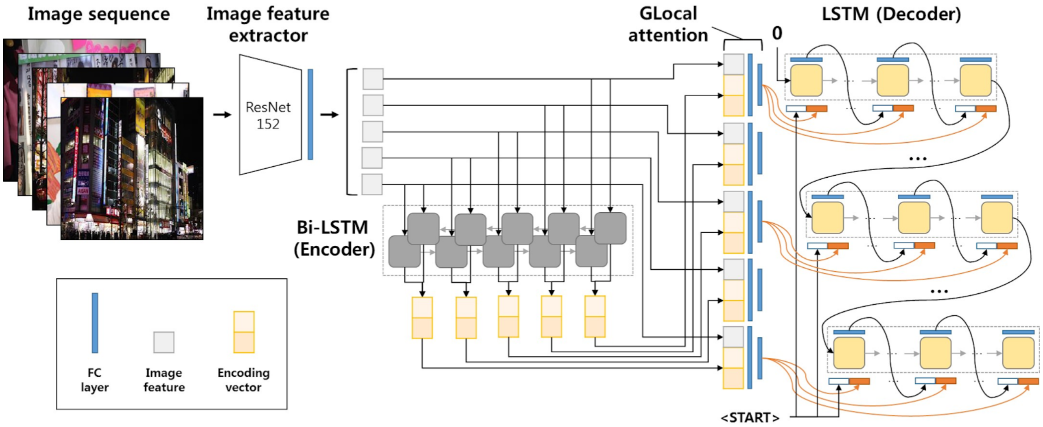


Figure 3.8: GLocal Attention Cascading Network architecture by [27].

even aggregation. These outputs were categorized as global information. The input to the encoder was 2048 dimensional vectors with hidden size of 1024 and dropout of 0.5. Owing to the large dimensionality of the encoder, batch normalization [23] was employed to deal with the co-variance shift among the extracted image features. Two layers of encoder were stacked, to potentially incorporate more nuances within the visual features.

Before cuing the decoder component, the global vectors (extracted from the bi-LSTM) are concatenated with the image feature vectors, otherwise termed local vectors. The resulting vectors were referred to as glocal vectors, and said to include both the overall and specific information of the sequence. These glocal vectors pass through fully connected layers to meet the dimensionality of the word embeddings. The decoder component is an LSTM which receives a combination of the glocal vector and the respective word representation at every timestep. The proposed cascading mechanism is using the same glocal vector till an end of sentence `<end>` token is obtained and working with the next glocal vector, till exhaustion. Initialized to zeros once, the hidden state of the previous sentence is cascaded on to the following sentence.

A set of generation phase heuristics were utilized to avoid repetition in the resulting sentences. At every timestep, the probability distribution of the language model (decoder) is sampled for one hundred times to create a pool of words. Word with maximum frequency is selected and simultaneously an additional cache holding the inverse counts of these sampled words is maintained, until the entire sentence-story is generated. This cache is utilized to decrease the sensitivity of choosing the already sampled words at later timesteps. Learning rate of 0.001 was used with Adam optimizer and a

Story (GLAC model): *[female] was so happy to see her grandson . she was happy with her . they had a lot of fun . everyone was having a great time . even the little ones enjoyed themselves .*

Story (human annotated): *when we finally brought our son home from the hospital so many people were at home with us to see him. everyone wanted a chance to hold him! we were all so happy to have a new addition to the family. my parents were so proud to be grand parents! i am so happy and i love my son very much!*

Figure 3.9: Image sequences and corresponding stories generated by the GLAC-net model [27].

batch size of 64. The dimensionality of the word embeddings was 256 and the hidden size of the language model LSTM was set to 1024. Teacher forcing mechanism of using the expected output at current timestep as input for the next timestep, was used for training the decoder. Batch normalization was also used in the decoder to avoid over-fitting. The model was trained for 100 epochs and it achieves a METEOR score of 0.3005 on the test split of the dataset. A sample story generated by their model is shown in Figure 3.9. The authors note that the stories are slightly monotonous. Upon verifying the model by training and testing, we emphasize the same. Additionally, we state that the trivial nature of the texts in the VIST dataset might also be one of the factors influencing the model to be presumptuous.

3.5 Adversarial Reward Learning

All the methods detailed in Sections 3.1 through 3.4 use the cross-entropy function to calculate the loss on the generated predictions. Section 2.1.1 explaining the decoder module, states that it outputs a probability distribution over the vocabulary of words. Specifically, a vector of probabilities at each timestep. The cross-entropy between the prediction vector $\hat{y}$ and the ground truth classes y is calculated as:

$$\mathtt{CE}_{\text{Loss}}(y, \hat{y}) = -\sum_{i} y_i \cdot \log(\hat{y}_i) \quad . \tag{3.1}$$

The words in the vocabulary are considered as classes and employing a multi-class log-loss function like cross-entropy for accessing the predictions has its implications. Fundamentally, the function rewards or punishes the model solely based on the probability of correct classes, by design. The loss value calculated is completely independent of the remaining probability split between the incorrect classes. Training mechanism based on such a criteria works very well for mutually exclusive multi-class classification scenarios like image classification [54]. However, for the use-case of visual storytelling or rather image captioning, the correctness of the output is subjective in nature. Therefore training a model based on cross entropy criterion which essentially tries to maximize the likelihood of the observed stories will yield a model suffering from exposure bias.

Another problem with modeling using the methods from Sections 3.1 through 3.4 is that the training and testing phases are asynchronous in terms of their driving objectives. Cross-entropy loss trains the model but NLP metrics like METEOR test the quality of the trained model. To address the problem of exposure bias and the state of asynchronicity an optimization approach called self-critical sequence training (SCST), based on reinforcement learning was proposed [51]. This was initially targeted towards regular image captioning systems and it is explained from the visual storytelling perspective in Section 4.1. In abbreviation, the method exposes the model to its own ongoing distribution and maximizes the expected return by optimizing its policy. Nevertheless, the self-critical loss criterion and related variations utilize a hand-crafted scorer (like METEOR) for rewarding and optimizing the model. Although this approach solves the issues with cross entropy, it brings with it the implicit limitations of automatic evaluation metrics which prevent the model to learn more intrinsic semantic details. Therefore hand-crafted methods are either too biased or too sparse to drive the search for optimal policy [62].

Addressing and detailing the above mentioned bottlenecks, the authors of *"No Metrics Are Perfect: Adversarial Reward Learning for Visual Storytelling"* [62] present an alternate approach. The proposed learning framework in Figure 3.10 is based on the paradigm of inverse reinforcement learning (IRL) in AI. Formally, the IRL approach is considered as a process of deriving a reward function from the observed expert (typically human) behavior. Unlike traditional RL, where reward functions need to be engineered, IRL learns the reward functions by prioritizing corresponding actions relying on a task oriented algorithm. Learning true reward functions is impractical owing to the exhaustive list of possible actions and thereby the authors resort to a neural network based reward model.

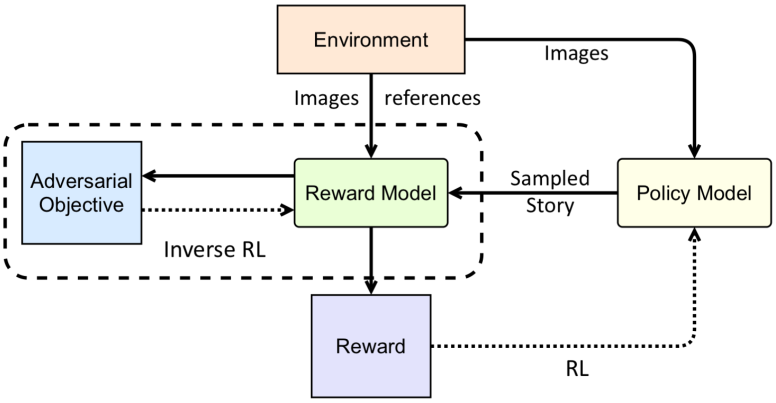


Figure 3.10: Adversarial Reward Learning framework [62].

The policy model presented in Figure 3.11 is a standard encoder-decoder style network which reads the image sequence as input and generates a story. There is a pretrained CNN feature extractor module and the authors use Resnet [18]. The visual encoder is a bi-directional RNN, specifically GRU. For the output of the encoder module at every timestep, there are RNN based decoder components with shared weights generating five sentence-stories making up for a story. The policy model was denoted as $\pi_\beta(W)$, where W being the word sequence of the story and β represents the model parameters. The reward model presented in Figure 3.12 reads the image sequence and the story as input and outputs a reward. It follows a CNN based architecture with different sized kernels with the motivation of extracting the `n`-grams of the story provided. The same pretrained CNN (from the policy model) is used for visually representing each image. Subsequently, max-pooling and fully connected layers are employed for projecting the visual and textual representations into a sentence-space. Rewards are calculated for each of the sentence stories separately with the intention of valuing fine-grained details. The reward model was denoted as $R_\theta(W)$, where W is the word sequence of the story and θ represents the model parameters.

Inspired by the min-max training strategy of generative adversarial networks, the authors propose an adversarial reward learning algorithm for training the policy and reward models. The objective for the reward model is:

$$\frac{\partial J_\theta}{\partial \theta} = \underset{W \sim p_e(W)}{\mathbb{E}} \left[\frac{\partial R_\theta(W)}{\partial \theta} \right] - \underset{W \sim \pi_\beta(W)}{\mathbb{E}} \left[\frac{\partial R_\theta(W)}{\partial \theta} \right] \tag{3.2}$$

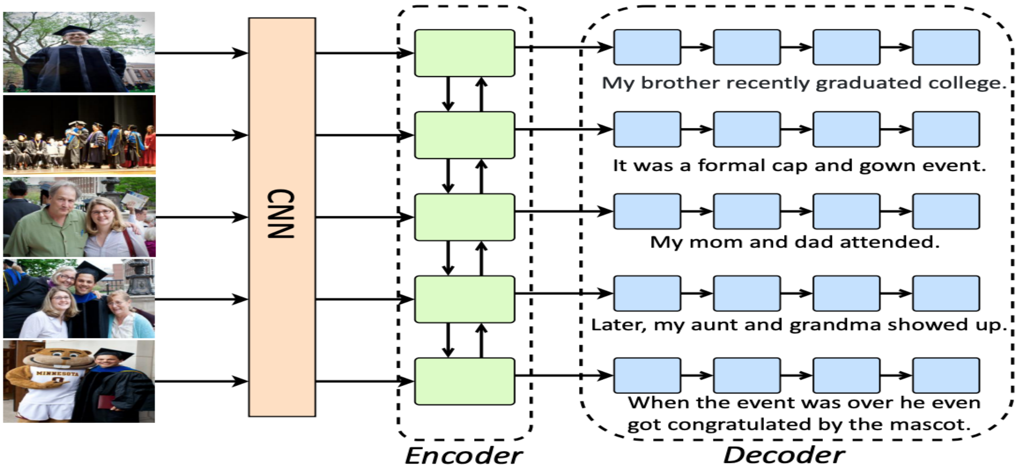

CNN
My brother recently graduated college.
It was a formal cap and gown event.
My mom and dad attended.
Later, my aunt and grandma showed up.
When the event was over he even got congratulated by the mascot.
Encoder
Decoder

Figure 3.11: AREL policy model [62].

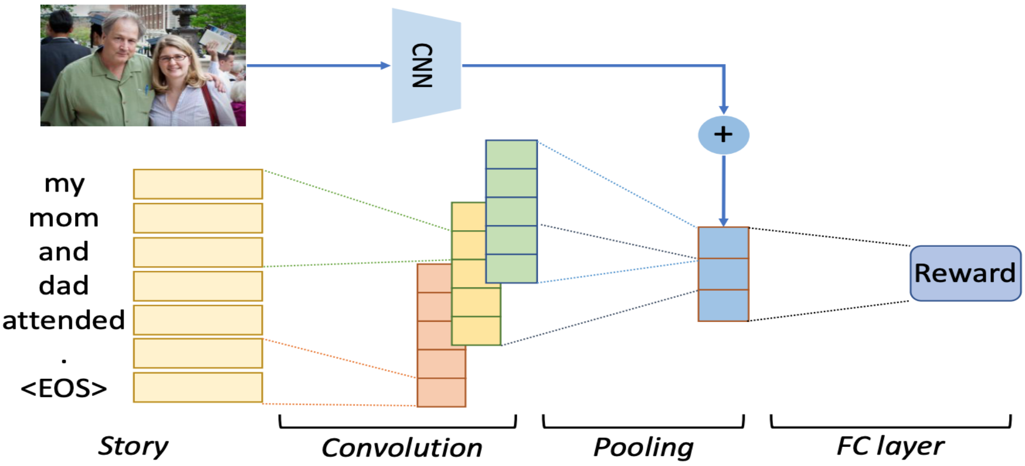
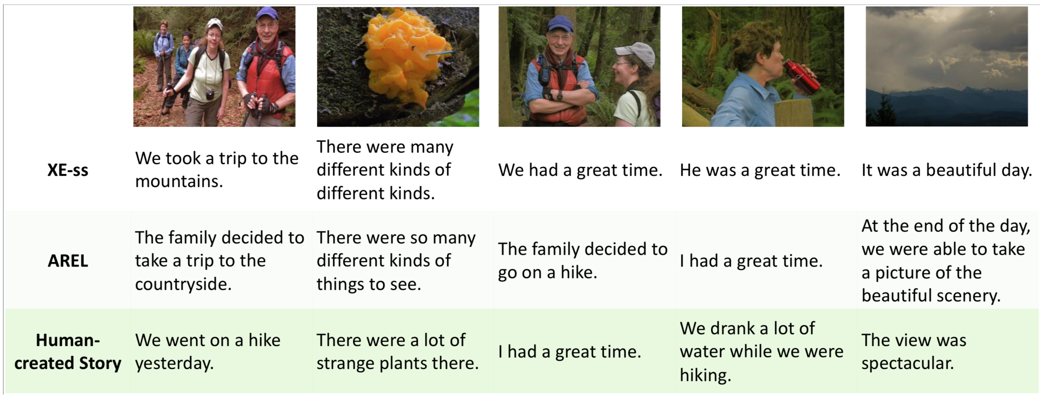


Figure 3.12: AREL reward model [62].

and for the policy model:

$$\frac{\partial J_\beta}{\partial \beta} = \mathop{\mathbb{E}}_{W \sim \pi_\beta(W)} \left[(R_\theta(W) - \log \pi_\beta(W) - b) \frac{\partial \log \pi_\beta(W)}{\partial \beta} \right] \quad . \tag{3.3}$$

After a standalone training of the policy model for the purpose of warming up, both the models are trained alternatively. The reward on a story generated by the policy model (in evaluation mode), combined with the reward on the respective ground truth story, is back-propagated as the loss to the reward model. The loss to the policy model is a combination of the reward for the generated story, the cross-entropy loss, and an estimated baseline function. The described purpose of this function is to account for variance among the timesteps and thereby within the batches. Practically, a linear parameter was learned to estimate the baseline values at every timestep of the policy model decoder. The model was trained for 100 epochs with a learning rate of 0.0004 and Adam optimization. GRU was used as the RNN cell with a hidden dimensionality of 512, a batch size of 64, and a dropout of 0.5 in the language model decoder. The sample results can be seen Figure 3.13. METEOR was used for calculating self-critical rewards and the reported score on the test split of the dataset was 0.35, which is state-of-the-art in visual storytelling. However, the authors emphasize human evaluation for evaluating aspects of relevance, expressiveness, and correctness of the generated stories.



XE-ss	We took a trip to the mountains.	There were many different kinds of different kinds.	We had a great time.	He was a great time.	It was a beautiful day.
AREL	The family decided to take a trip to the countryside.	There were so many different kinds of things to see.	The family decided to go on a hike.	I had a great time.	At the end of the day, we were able to take a picture of the beautiful scenery.
Human-created Story	We went on a hike yesterday.	There were a lot of strange plants there.	I had a great time.	We drank a lot of water while we were hiking.	The view was spectacular.

Figure 3.13: AREL results and comparison with baseline cross-entropy model (XE-ss) [62].

3.6 Hierarchically Structured Reinforcement Learning for Topically Coherent Visual Story Generation

Upon considering the challenges and limitations of the approaches in Sections 3.2 through 3.5 the authors of the *Visual Storytelling* [22] task propose a hierarchical model based approach. They introduce a framework comprising a two-level hierarchical decoder. The high-level decoder generates a semantic topic for each image in the sequence and the low-level decoder generates a sentence for each image based on the topics, using a semantic compositional network [13] based language model. Reinforcement learning is used to train the two decoders jointly, which is closely related to the adversarial reward learning approach, detailed in Section 3.5. However, the authors differentiate their method through the novelty of exploring a "plan-ahead" strategy by using the topics generated by the high-level decoder. Similar to the AREL [62] work, the authors of this method criticize the use of maximum likelihood estimation based training criteria and opine that MLE will fail to exploit the information wealth across the long span of stories. The proposed framework is shown in Figure 3.14.

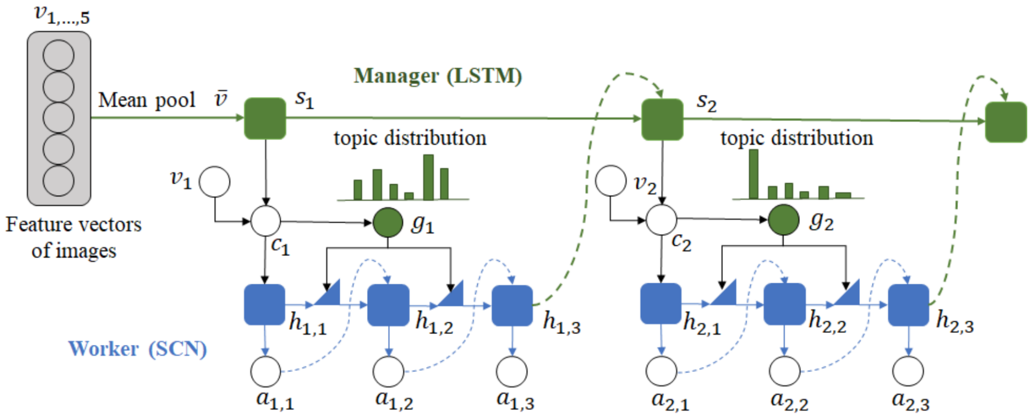


Figure 3.14: Proposed hierarchical framework by [21].

The encoder component is different compared to all the other methods discussed in this chapter. After extracting the feature vectors of the images in a sequence, using a pretrained CNN, mean pooling was applied to obtain an image-sequence context vector $\bar{v}$. Although specifics about pooling were unavailable, one can assume that the five feature vectors were sequentially merged and a 1D mean pooling was performed on the resulting representation.

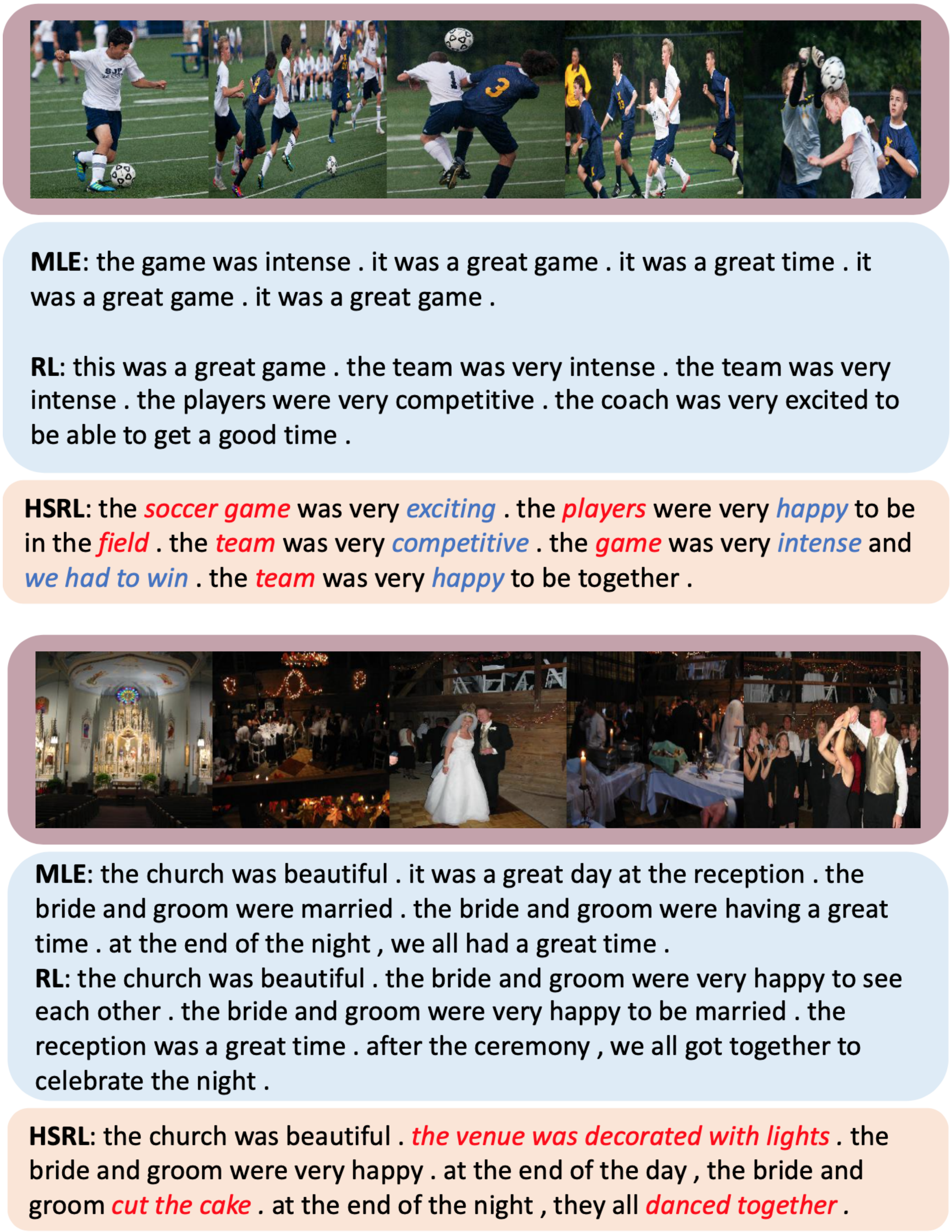
The authors use the terms *manager* add *worker* for the high and low level decoders, respectively. In essence, the objective of the manager module is to generate a topic distribution g_l and a context vector c_l by reading the corresponding image feature and the last available hidden state of the worker module, depending on the mode of training. The worker module is an semantic compositional network (SCN) decoder which reads both the context and the topic distribution vector for generating sentences. The objective for the manager module is:

$$\mathcal{L}_{\text{mle}}^{M}\left(\theta_m\right) = -\sum_{\ell=1}^{n} \log p_{\theta_m}\left(g_\ell^* | g_1^*, \ldots, g_{\ell-1}^*, h_{l-1,T}\right) \tag{3.4}$$

and for the worker module:

$$\begin{aligned}
\mathcal{L}_{\text{mle}}^{W}\left(\theta_w\right) &= -\sum_{\ell=1}^{n}\sum_{t=1}^{T} \log p_{\theta_w}\left(y_{\ell,t}^* | y_{\ell,1}^*, \ldots, y_{\ell,t-1}^*, g_\ell, c_\ell\right) \\
\mathcal{L}_{\text{rl}}^{W}\left(\theta_w\right) &= -\left(r^\star - \hat{r}\right)\sum_{\ell=1}^{n}\sum_{t=1}^{T} \log p_{\theta_w}\left(\hat{y}_{\ell,t} | \hat{y}_{\ell,1}, \ldots, \hat{y}_{\ell,t-1}, g_\ell, c_\ell\right) \\
\mathcal{L}_{\text{mix}}^{W} &= \gamma\mathcal{L}_{\text{rl}}^{W} + (1-\gamma)\mathcal{L}_{\text{mle}}^{W} \quad .
\end{aligned} \tag{3.5}$$

In the above expressions, n denotes the number of topic sequences and T denotes the number of words. The $\mathcal{L}_{\text{mle}}^{M}\left(\theta_m\right)$ minimizes the negative log likelihood of predicting the next topic in the story, given the ground truth set of topics. The $\mathcal{L}_{\text{mix}}^{W}$ is a combination of log likelihood and self critical sequence loss on the generated sentence with respect to the ground truth. The intention of such a mixture is to encourage the model to generate sentences for receiving more reward rather than merely greedily copying the true stories. The work explores various ways of learning or training the modules together, i.e., cascaded training, iterative training and joint training. The experimental results reported claim that learning through joint training of the manager and worker produces the state-of-the-art METEOR score of 0.3523 on the VIST dataset (the split was not mentioned). Sample results reported by the paper are shown in Figure 3.15.



MLE: the game was intense . it was a great game . it was a great time . it was a great game . it was a great game .

RL: this was a great game . the team was very intense . the team was very intense . the players were very competitive . the coach was very excited to be able to get a good time .

HSRL: the *soccer game* was very *exciting* . the *players* were very *happy* to be in the *field* . the *team* was very *competitive* . the *game* was very *intense* and *we had to win* . the *team* was very *happy* to be together .

MLE: the church was beautiful . it was a great day at the reception . the bride and groom were married . the bride and groom were having a great time . at the end of the night , we all had a great time .

RL: the church was beautiful . the bride and groom were very happy to see each other . the bride and groom were very happy to be married . the reception was a great time . after the ceremony , we all got together to celebrate the night .

HSRL: the church was beautiful . *the venue was decorated with lights* . the bride and groom were very happy . at the end of the day , the bride and groom *cut the cake* . at the end of the night , they all *danced together* .

Figure 3.15: Results reported by the topically coherent manager-worker model [21].

Chapter 4

Remodeling

From Chapter 3, it is evident that amidst a lot of common styles in architecture, there are notable differences in terms of several moving parts of the visual storytelling models. In this chapter various components and aspects of selected models from Chapter 3 are remodeled with the intention of analyzing behaviors and detecting patterns. The choice of these hybrid setups is heavily intuition driven. Empirically, other experiments with several other combinations were performed and this chapter reports two of the most promising scenarios in the following sections. Evaluation scores are summarized in Table 4.1 at the end of this chapter together with results from Chapter 3.

4.1 GLAC Net with SCST

Exposure bias in the sequence-to-sequence paradigm and particularly in the task of captioning is a severe bottleneck. Formally, this problem occurs due to the conditioning of the language model on ground truth rather than generated words or sentences. Employing the cross-entropy loss provided under equation (3.1) for learning causes a mismatch between the training and testing phases. While the model is trained to produce words adhering to maximizing the likelihood of the ground truth, during inference it is evaluated and scored subjectively using NLP metrics like METEOR. This discrepancy causes the model to stumble and generate sentences with hitches like repetition and sparse vocabulary. Some early enhancements like scheduled sampling [5] and professor forcing [33] were proposed to address this issue. Scheduled sampling tries to bridge the gap between training and inference by balancing the usage of true previous tokens y_{t-1} and estimated previous tokens $\hat{y}_{t-1}$ during training. Professor forcing employs an additional discriminator network and expects the language-generating-model to switch between teacher-forcing [63] and non-teacher-forcing modes while learning to fool the discriminator. However, the above-mentioned techniques only succeed to an extent.

Meanwhile, RL-based techniques started to show tremendous progress in various modeling facets. Although RL and particularly policy gradient methods were around for years, [2] was the first of works to apply them for supervised learning-based sequence prediction. From the stance of captioning, [51] is the work that formalized the commonalities and paved the way for RL-based supervised learning mechanisms. The work defines image captioning from the RL Markov Decision Process (MDP) setting. The work adapted architecture detailed under Section 2.1.1 considering the model as *agent* and the visual features along with previous words $W^s_{1:t-1}$ as the *environment*. Generation of the next word W^s_t is an *action* given the models' *state* and its associated probability distribution or *policy*. *Reward* $r(W^s)$ is computed at the end of each episode, which in the case of captioning is following the generation of the `<EOS>` token. Superficially, RL methods are either value-based or policy-based. In the setting of captioning, practically, policy-based methods which target to optimize the policy (probability distribution) of the model directly, suit well. However, neural nets (supervised learning) learn through gradients propagated as feedback. Policy-gradient algorithms attempt to optimize the loss function $L(\theta)$ with the objective of minimizing the negative expected reward:

$$L(\theta) = -\mathbb{E}_{W^s \sim P_\theta}[r(W^s)] \quad . \tag{4.1}$$

Restating the motivation, using cross-entropy loss during training and a natural-language-processing (NLP) metric for scoring the inference sentences was a mismatch. Therefore the loss in equation (4.1) scores the generated sentences during training using the same NLP metric $r(W^s)$ and intends to minimize the expected negative reward. Emphasizing that the reward function is non-differentiable, the REINFORCE algorithm [58] derives and defines the gradient to be propagated with respect to the model parameters θ in equations (4.2) through (4.11) as follows:

$$\nabla_\theta L(\theta) = -\nabla_\theta \mathbb{E}_{W^s \sim P_\theta}[r(W^s)] \quad . \tag{4.2}$$

Considering captioning is a discrete task, the expectation $\mathbb{E}$ of the reward would be a summation over rewards for all possible sentences W_s, weighted by their probabilities $P_\theta(W^s)$, that can be sampled from the model:

$$\nabla_\theta L(\theta) = -\nabla_\theta \sum_{W^s} r(W^s) P_\theta(W^s) \quad . \tag{4.3}$$

The score-function trick from [58], i.e., multiplying and dividing by the probability term gives:

$$\nabla_\theta L(\theta) = -\sum_{W^s} r(W^s)\nabla_\theta P_\theta(W^s) = -\sum_{W^s} r(W^s)\frac{\nabla_\theta P_\theta(W^s)}{P_\theta(W^s)}P_\theta(W^s) \quad . \tag{4.4}$$

Applying the fact about derivatives of log functions:

$$\nabla_x \log f(x) = \frac{f'(x)}{f(x)} \tag{4.5}$$

to equation (4.4) yields:

$$\nabla_\theta L(\theta) = -\sum_{W^s} r(W^s)P_\theta(W^s)\nabla_\theta \log P_\theta(W^s) \quad . \tag{4.6}$$

Converting equation (4.6) back into the expectation term using the above-mentioned definition, leads to a formulation that is directly mentioned in the SCST paper [51]:

$$\nabla_\theta L(\theta) = -\mathbb{E}[r(W^s)\nabla_\theta \log P_\theta(W^s)] \quad . \tag{4.7}$$

However, in practice, the research community approximates the expected gradient in equation (4.7) by using a single sample W^s for each mini-batch during training. Therefore using W^s, the REINFORCE expression for gradient is:

$$\nabla_\theta L(\theta) \approx -r(W^s)\nabla_\theta \log P_\theta(W^s) \quad . \tag{4.8}$$

Equation (4.8) is a general expression with respect to the model parameters θ. Nevertheless, language-generating networks in captioning contain Softmax as a final layer to convert logits $\texttt{logit}_{1:T}$ into probabilities $P_\theta(W^s_{1...T})$ as:

$$\text{Softmax}(\texttt{logit}_{1:T}) = P_\theta(W^s_{1:T}) = \frac{\exp^{\texttt{logit}_i}}{\sum_j \exp^{\texttt{logit}_j}}, \forall i \in \{1, \ldots, T\} \quad . \tag{4.9}$$

Because computation of the gradients and back-propagation follows the chain rule, computing the gradients with respect to the model parameters θ requires the gradients to be computed from the Softmax layer. This statement can be formulated mathematically as:

$$\frac{\partial L(\theta)}{\partial \theta} = \sum_{t=1}^{T} \frac{\partial L(\theta)}{\partial \texttt{logit}_t} \frac{\partial \texttt{logit}_t}{\partial \theta}, \tag{4.10}$$

and we intend to obtain the gradient only for one single timestep t. Therefore, we first derive $\frac{\partial \text{Softmax}(\texttt{logit}_i)}{\partial \texttt{logit}_t}$ as:

$$\begin{aligned}
\frac{\partial \text{Softmax}(\texttt{logit}_i)}{\partial \texttt{logit}_t} &= \frac{\partial \dfrac{\exp^{\texttt{logit}_i}}{\sum_j \exp^{\texttt{logit}_j}}}{\partial \texttt{logit}_t}, \\
&\implies \text{applying quotient rule}^{1} \\
&= \frac{\exp^{\texttt{logit}_t} \sum_j \exp^{\texttt{logit}_j} - \exp^{\texttt{logit}_t} \exp^{\texttt{logit}_i}}{(\sum_j \exp^{\texttt{logit}_j})^2}, \\
&\implies \text{generalizing for } t = i \text{ and } t \neq i \qquad (4.11) \\
&= \text{Softmax}(\texttt{logit}_t)(1_{ti} - \text{Softmax}(\texttt{logit}_i)), \\
&\implies \text{from equation (4.9)} \\
&= P_\theta(W_t^s)(1_{ti} - P_\theta(W_i^s))), \\
&\implies \text{plugging into the gradient of } L(\theta) \quad ,
\end{aligned}$$

and finally plug the result into $\frac{\partial L(\theta)}{\partial \texttt{logit}_t}$ as:

$$\frac{\partial L(\theta)}{\partial \texttt{logit}_t} = r(W^s)(P_\theta(W_t^s) - 1_{W_t^s}) \quad . \tag{4.12}$$

Nonetheless, the use of REINFORCE empirically indicates that the acquired gradient estimates can have high variance, owing to the inherent leniency of approximating using one single sample for the entire mini-batch. To tackle the aforementioned obstacle, [49] uses a baseline reward b, based on which the gradients either turn out positive or negative depending on the score of the sampled sequence with respect to a baseline. Choosing a valid baseline is challenging and several proposals suggest learning it as a parameter during training [49], [58]. In [51] it was proposed to utilize the reward of a sequence

[1] https://en.wikipedia.org/wiki/Quotient_rule

obtained through greedy sampling of the model as a baseline. The intuition is to only provide a positive reward to generated sequences (predicted training samples) which are better than the current model output (greedy sample). This way the authors claim guaranteed model improvements. An additional advantage is to avoid learning separate baseline functions. This section applies self-critical learning approach to the model architecture of [27] detailed under Section 3.4. Experiments, results and comparisons with the cross-entropy based learning are reported in the following subsection.

4.1.1 Experiments and results

For experiments using SCST, the original GLAC-net settings are maintained to ascertain comparability. The model comprises a bi-directional LSTM network which reads the image features obtained from Resnet-152 [18], sequentially. The glocal vectors are made from the concatenation of encoder outputs at each timestep with respective image features. The decoder reads the glocal vectors individually and produces sentences that make up for a story. The `<end>` token marks the completion of exploiting each glocal feature vector of the sequence, by the decoder. Before training, the images of VIST dataset are resized to have a resolution of 256, and during training, the images are horizontally flipped, cropped randomly with a resolution of 224 and normalized to have pixel values between 0 and 1. For optimization, a learning rate of 0.0001 with Adam optimization and weight decay of 0.00001 are employed. Embedding layer is learned for representing each word with 256-dimensional vectors. The model is warmed-up for 85 epochs with a batch size of 64 and cross-entropy criterion before considering the SCST objective for 15 more epochs. The SCST loss is considered in 1 : 1 ratio with the regular cross-entropy loss during training. METEOR scoring on the VIST dataset test split across the epochs is visualized with and without using SCST objective in Figure 4.1. Additionally, the batch level advantage between epochs is presented in Figure 4.2. Sample stories generated by the model are provided in Figure 4.3. Assessment of the model behavior and quality of the generated results are discussed in Chapter 6.

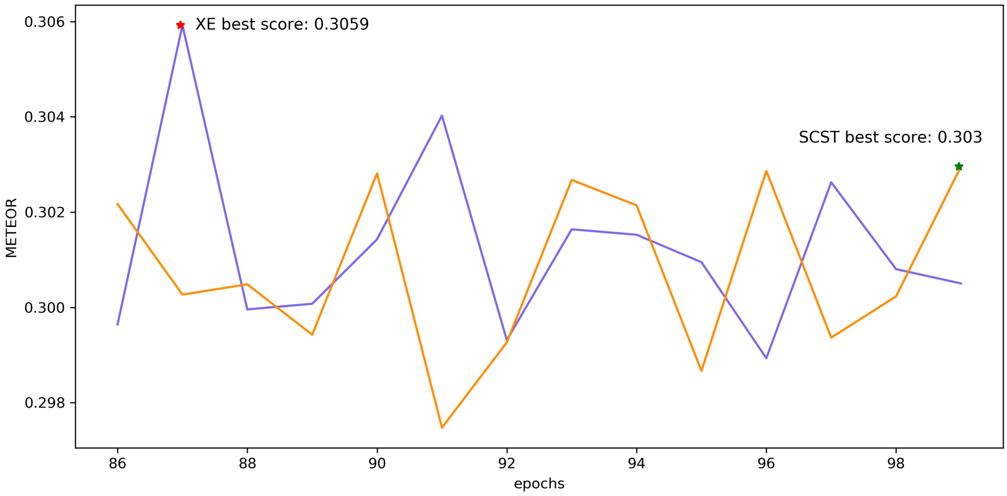


Figure 4.1: Visualization of METEOR scores across epochs for GLAC-net under both cross-entropy and SCST objectives.

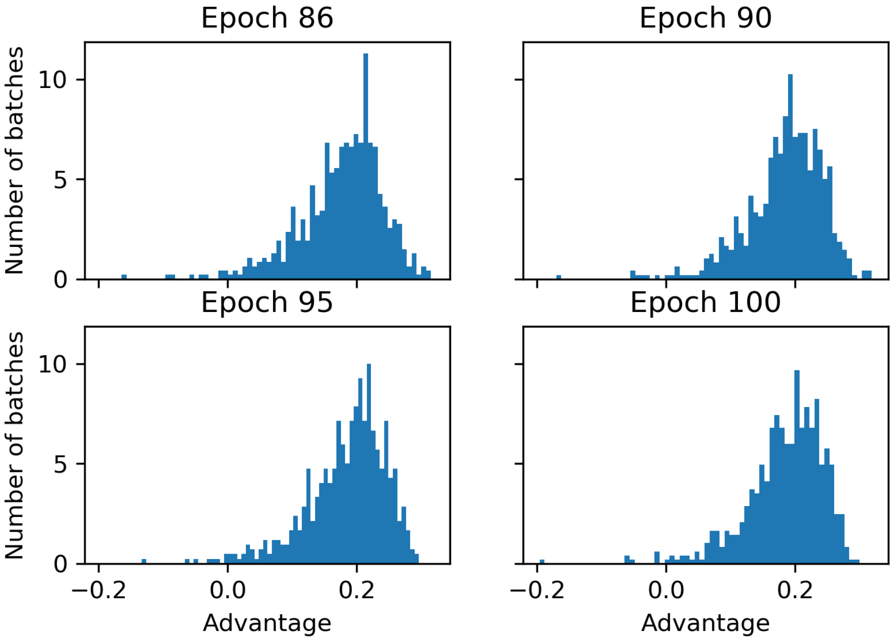
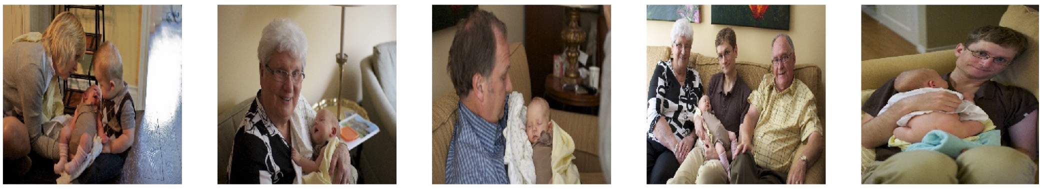


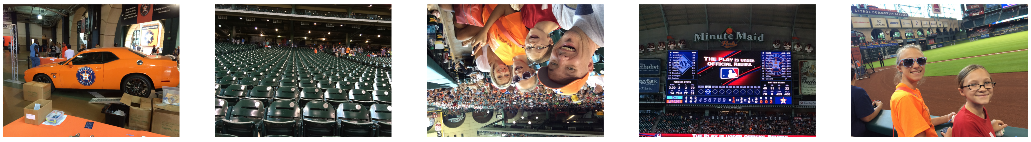
Figure 4.2: Advantage over the baseline among mini-batches during the learning.



Story (GLAC-SCST model): *the family was excited for their baby 's birthday . the boy was so happy to be able to see his new grandson . the grandparents were very happy . the younger brother was playing with his new toy . the baby was having a great time .*

Story (GLAC-XE model): *[female] was so happy to see her grandson . she was happy with her . they had a lot of fun . everyone was having a great time . even the little ones enjoyed themselves .*

Story (human annotated): *when we finally brought our son home from the hospital so many people were at home with us to see him. everyone wanted a chance to hold him! we were all so happy to have a new addition to the family. my parents were so proud to be grand parents! i am so happy and i love my son very much!*



Story (GLAC-SCST model): *the stadium was packed . the seats were empty . the fans were excited . the game was very exciting . the game was a lot of fun .*

Story (GLAC-XE model): *the car was very fast . the seats were empty . the traffic was good . the stadium was not crowded . the fans were excited .*

Story (human annotated): *i took the family to a baseball game and we saw this awesome car before the game , that you had a chance to win in a raffle . we were lucky enough to get to the game early and our seats were amazing . of course we had to get a family selfie during the game . the scoreboard was so huge , i had to get a photo of it . the girls loved being so close to the field that they could reach out and touch the ground if they wanted to .*

Figure 4.3: Stories generated by GLAC-net model under SCST learning compared with MLE based training (GLAC-XE).

4.2 AREL framework with GAN objective

The authors of the AREL [62] work argued against the aspect of likelihood estimation based methods for visual storytelling, which was discussed in Section 3.5. A major highlighted flaw is the exposure bias problem. During the training process, the model learns to generate text sequentially and predicts the next token based on the already predicted words. During inference, the model generates sentences that might not necessarily be part of the training data. Therefore, using likelihood estimation methods like cross-entropy as objectives to train the model does not adhere to the expectation of language generation tasks like captioning. Moving away from MLE, the community has adapted RL based policy gradient methods for standard captioning and thereby visual storytelling. Section 4.1 provided specifics about those approaches. However, the approaches using NLP metrics for policy search bring with them a different difficulty of choosing the automatic metric for guiding and rewarding the model.

All the automatic evaluation measures are hand-crafted to perform some form of string-based matching between the generated sentence and the ground truth. Nevertheless, they do not correlate well with the human judgment of quality and coherency, particularly in the case of long stories.

> *We had a great time to have a lot of the. They were to be a of the. They were to be in the. The and it were to be the. The, and it were to be the.*

is one example story which achieves a high METEOR score of 0.402, but is mostly incomprehensible [62]. Additionally, these measures are not heavily inter-correlated either. Although they share some underlying steps, their assessment of the hypothesis text is rather distinctive. Consequently, they fail to drive policy-search for a language model.

A new direction towards addressing these bottlenecks is employing adversarial frameworks. Traditionally, generative adversarial networks served the purpose of learning continuous data distributions. However, text generation is a discrete problem and standard language models do not allow for propagation of discontinuous gradients. So employing the adversarial pipeline naturally calls for RL based policy-gradient estimation techniques. The initial breakthrough towards such variation is SeqGAN [66]. The authors of the AREL model heavily rely on the idea of sequence GAN [66], presenting reward and policy models as *discriminator* and *generator*, respectively. They compare their model against a standard GAN model and report results. This section reports experiments using the standard GAN objective and compares the results. The gradient computation for the generator model is:

$$\frac{\partial J_\beta}{\partial \beta} = \underset{W \sim \pi_\beta(W)}{\mathbb{E}} [R_\theta(W)] \quad , \tag{4.13}$$

and for the discriminator model is as follows:

$$\frac{\partial J_\theta}{\partial \theta} = \underset{W \sim p_e(W)}{\mathbb{E}} \left[\frac{\partial R_\theta(W)}{\partial \theta} \right] - \underset{W \sim \pi_\beta(W)}{\mathbb{E}} \left[\frac{\partial R_\theta(W)}{\partial \theta} \right] \quad . \tag{4.14}$$

4.2.1 Experiments and results

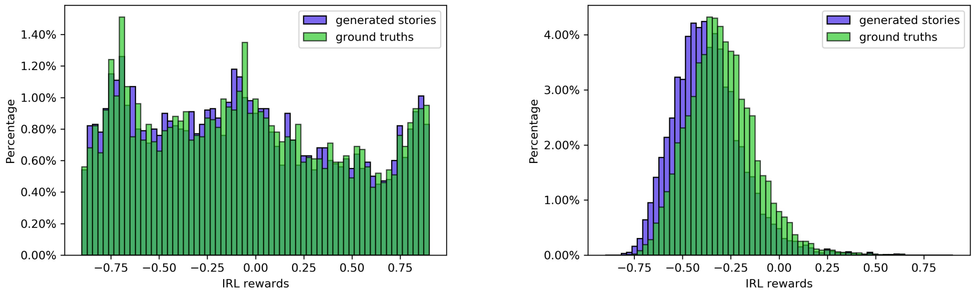
This section reports experiments performed employing the above-mentioned objectives for the policy and reward models. The VIST dataset described under Section 2.3 is utilized for both the training and evaluation phases. As a preprocessing step, the policy model is trained for 100 epochs using the cross-entropy objective for the purpose of stabilizing the learning process. Similar to the other experiments, a pretrained Resnet-152 [18] model is used for extracting the visual features of the five images per sequence. The vocabulary/dictionary of words is built to include words appearing three or more times in the training text corpus yielding a size of 9,837. Upon obtaining the image sequence features from the CNN, the visual encoder module of the policy model, which is a bi-directional GRU RNN with 256 hidden units for each direction, represents them. The hidden states in both directions of the encoder are concatenated and the module learns to incorporate context into these 512-dimensional vectors.

The overall image sequence subject and bi-directional context obtained at the five timesteps of the encoder are passed on to the decoder module. Authors of AREL employ five parallel decoders with shared weights. Each decoder is a unidirectional GRU RNN that generates sentences making up for a story. Scheduled sampling [5] with a threshold probability of 0.25 is employed for training the decoders. A uniform distribution $\mathrm{U}(0, 1)$ is sampled per mini-batch to decide between choosing the estimated and true previous token. Adhering to the same setup, a consolidated story from the policy model $W \sim \pi_\beta(W)$ is then provided to the reward model for obtaining the reward $R_\theta(W)$.

The reward model is a CNN based design as shown in Figure 3.12 to extract n-gram features of a sentence. For this purpose, convolution kernels f_{conv} of sizes 2, 3 and 4 with stride 1 are employed. These kernels stride through the sentence representations incorporating uni/bi/tri-gram abstractions. One-dimensional max-pooling is applied on these extractions before concatenating with the respective visual feature representation of the respective image from the sequence, obtained using Resnet-152 [18]. Embedding layer is learned for representing words in the policy and reward networks. The word embedding vectors of the reward model and the CNN filters are 128 dimensional. Eventually, the reward is computed as a linear projection W_r of the combined vector with soft-sign non-linearity applied:

$$\begin{aligned} R_\theta(W) &= \text{softsign}(W_r(f_{conv}(W) + W_i I_{CNN}) + b_r), \text{where} \\ \text{softsign}(z) &= \frac{z}{1+|z|} \quad . \end{aligned} \tag{4.15}$$

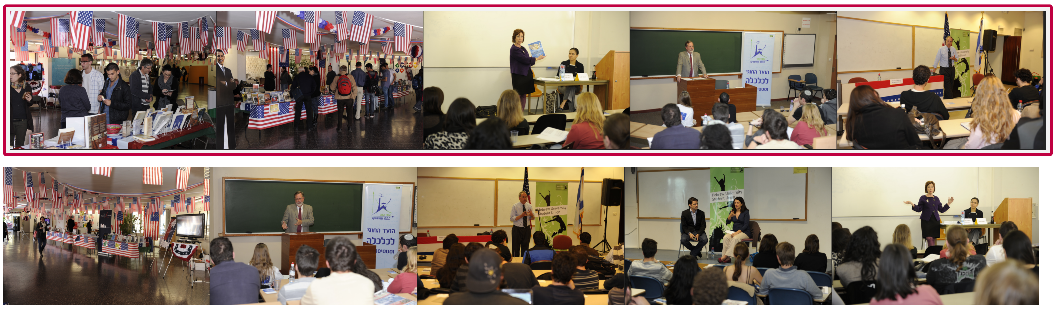
The alternating/cascading training strategy of the AREL model is also followed for our experiments with the GAN model. During the training phase, a batch size of 64, a learning rate of $2 \cdot 10^{-4}$ with Adam optimizer, a dropout of 0.5 for the reward model and a visual dropout of 0.2 for the policy model visual encoder module are used. For generating sentences during inference a beam size of 3 was used to search the probability space. Learned rewards upon experimenting with both the AREL and GAN objectives are visualized in Figure 4.4. Sample results produced by the model during inference are provided in Figures 4.5 and 4.6. For the purpose of scoring, AREL [62] combines all the stories belonging to the same album as a single reference against the generated hypothesis, and we abide by this process also for the GAN model scoring. The model achieves a METEOR score of 0.3540 on the VIST dataset test split. Evaluation measures on the stories generated by other models for the VIST test split are summarized in Table 4.1. Assessment of the model behavior and quality of the generated results are discussed in Chapter 6.



(a) AREL model

(b) GAN model

Figure 4.4: Comparison of learned rewards visualization of ground truths and generated stories by AREL and GAN models.

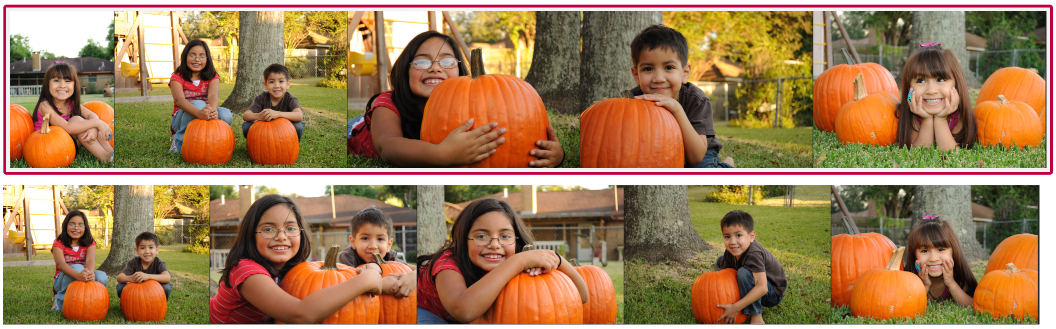


Story (GAN model): *there were a lot of people at the convention today . everyone was there to support the event . the speaker gave a speech about the students . the speaker gave a speech . after the presentation , the speaker gave a speech to the audience .*

Story (AREL model): *there were a lot of people at the convention . there were a lot of people there . there were many speakers there . there were a lot of people there . the presentation was a success , and everyone had a great time .*

Overall story (human annotated): *there was a huge american day at school today . everyone came to visit . even parents decided to show up ! ms. [female] was able to teach some adults the proper way to be an american . while mr. [male] discussed his value of being american . it was very nice to have mr. chang give us the final announcements for [female] day . it was career day , how exciting . a man gave a speech . then another man gave a speech . then a man and a woman gave a speech . then two women gave a speech ., he was career day at my college . people showed up from different companies to set up an informative booths . you got to listen to guest lecturers talk about their companies . the whole day was very informative and i managed to enjoy myself . i think i have finally found the career i want to be in . the conference held a book store as well . you can purchase patriotic books . some of the authors were present . the speakers took turns talking about issues . they took questions afterwards . the people who work for the county recorder's are a dedicated bunch and very patriotic . they go to classrooms and educate people on the importance of voting and on the history of voting in location, but there most important job is to educate the volunteers for the polling places .*

Figure 4.5: An image sequence (boxed in purple) from the VIST test split for which stories produced by the models are compared against each other.



Story (GAN model): *my sister and i went to a halloween party . the kids are having a great time . my sister and i got to see the pumpkins . my son was a little scary and i think it was a little scary . my sister and my sister and my brother , [male] , and [male] 's a great time .*

Story (AREL model): *the pumpkins are ready for halloween . [female] and [female] were all dressed up and ready for the party . my sister and her favorite pumpkin was a little scary . the little girl was having a great time . [female] and her friends are having a great time and had a great time .*

Overall story (human annotated): *the whole family went out to the pumpkin patch together . the brother and sister chose one pumpkin each . the sister was so happy with hers ! the brother also loved his pumpkin . the other sister was happiest of all , though , because she got four pumpkins . halloween is her favorite holiday . this boy and girl picked the pumpkins they wanted to carve . she said she is going to carve a princess on her pumpkin . he wants to carve batman on his pumpkin . she can't decide which pumpkin she wants to take home . can i take them all ,i got to pick out my first pumpkin to carve , today ! my sister and brother were there , too . i really think my sister got the biggest one out of all of us ! my brother insists his is the best , though . i 'm just happy that i had fun today , and i want to do it again , sometime !, it 's been two weeks since the pumpkins first appeared , the kids do n't seem to know any better . these pumpkins have murdered over 40 people nation wide , it 's good that the kids do n't know we 're being held hostage . this pumpkin could snap at any second and cut off my daughters head , i almost vomited after this picture . look how happy he is , he does n't know the damage this pumpkin has caused . it 's like the pumpkins are mocking me , they know how much these pictures hurt my soul .,the family went to a pumpkin patch . they had tons of fun with pumpkins . they loved how big and round they were . even the youngest got in on the fun . they loved their time at the patch .*

Figure 4.6: An image sequence (boxed in purple) from the VIST test split for which stories produced by the models are compared against each other.

4.3 Summary of scores

This section consolidates and summarizes the NLP evaluation scores obtained by several models for the visual storytelling task. The Baseline model is evaluated on validation split of the VIST dataset, considering the relevance when comparing with the respective reported scores. All the other models are evaluated on the test split of the VIST dataset. In the table below, "replicated" inside the parentheses indicate that experiments are performed and validated for this thesis and "reported" implies that the scores are taken from respective published papers.

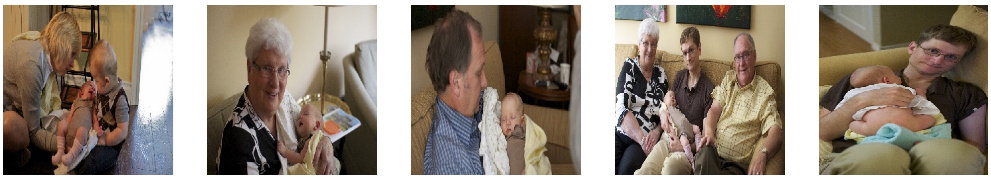
	Section	METEOR v 1.5	BLEU-4
Baseline model [22] (reported)	—	0.3142	—
Baseline model [22] (replicated)	3.1	0.2160	—
Multi encoder model [57] (reported)	—	0.2390	—
Multi encoder model [57] (replicated)	3.2	0.2250	—
Multi decoder model [15] (replicated and reported)	3.3	0.3400	—
GLAC-net model [27] (reported)	—	0.3014	—
GLAC-net model [27] (replicated)	3.4	0.3005	—
AREL model [62] (reported)	—	0.3500	**0.1410**
AREL model [62] (replicated)	3.5	0.3482	0.1354
Topically coherent manager-worker model [21] (reported)	3.6	0.3523	0.1232
GLAC-SC-net model (remodeled)	4.1	0.3029	—
GAN model (remodeled)	4.2	**0.3540**	0.1409

Table 4.1: NLP metrics detailed under Section 2.4, computed and compared for stories generated by models explained in Chapters 3 and 4.

Chapter 5

Character-centric storytelling

Although existing visual storytelling models generate narratives that read subjectively well, there could be cases when these models miss out on generating stories that account for and address all prospective human and animal characters in the image sequences. An example case is presented in Figure 5.1. Considering this scenario, we propose a model that implicitly learns relationships between provided characters and thereby generates stories with respective characters in scope. We use the VIST dataset for this purpose and report numerous statistics on the dataset. Eventually, we describe the model, explain the experiment, and discuss our current status and future work.



Story (GLAC model): *[female] was so happy to see her grandson . she was happy with her . they had a lot of fun . everyone was having a great time . even the little ones enjoyed themselves .*

Story (human annotated): *when we finally brought our son home from the hospital so many people were at home with us to see him. everyone wanted a chance to hold him! we were all so happy to have a new addition to the family . my parents were so proud to be grand parents! i am so happy and i love my son very much!*

Figure 5.1: An image sequence and corresponding story generated by [27] that lacks many prospective characters.

5.1 Data analysis

We used the VIST dataset comprising of image sequences obtained from Flickr albums and respective annotated descriptions collected through Amazon Mechanical Turk. More details about the composition and collection process of the dataset are under Section 2.3.

5.1.1 Character extraction

We extracted characters out of the VIST dataset. To this end, we considered that a character is either “a person” or “an animal.” We decided that the best way to do this would be by making use of the human-annotated text instead of images for the sake of being diverse (e.g., detection on images would yield “person”, as opposed to father).

The extraction takes place as a two-step process as shown in Figure 5.2.

1. **Identification of nouns**: We first used a pre-trained part-of-speech tagger [39] to identify all kinds of nouns in the annotations. Specifically, these noun categories are NN – *common, singular or mass*, NNS – *noun, common, plural*, NNP – *noun, proper, singular*, and NNPS – *noun, proper, plural.*

2. **Filtering for hypernyms**: WordNet [42] is a lexical database over the English language containing various semantic relations and synonym sets. A hypernym is one such semantic relation constituting a category into which words with more specific meanings fall. From among the extracted nouns, we thereby filtered those words that have their lowest common hypernym as either “person” or “animal.”

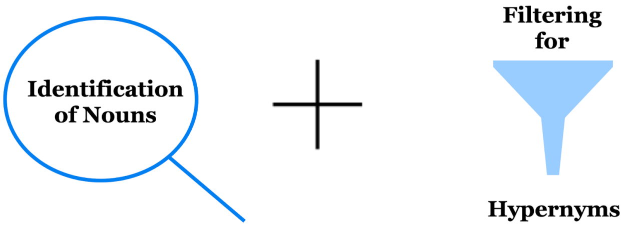


Figure 5.2: Character extraction process.

5.1.2 Character analysis

We analyzed the VIST dataset from the perspective of the extracted characters and observed that 20405 training, 2349 validation, and 2768 testing data samples have at least one character present among their stories. Approximately 50% of the data samples in the entire dataset satisfy this condition. To pursue the prominence of relationships between these characters, we analyzed these extractions for both individual and co-occurrence frequencies. We found a total of 1470 distinct characters with 1333 in training, 387 in the validation, and 466 in the testing splits. These numbers can be considered as an indication of the limited size of the dataset because the number of distinct characters within each split is strongly dependent on the respective size of that split.

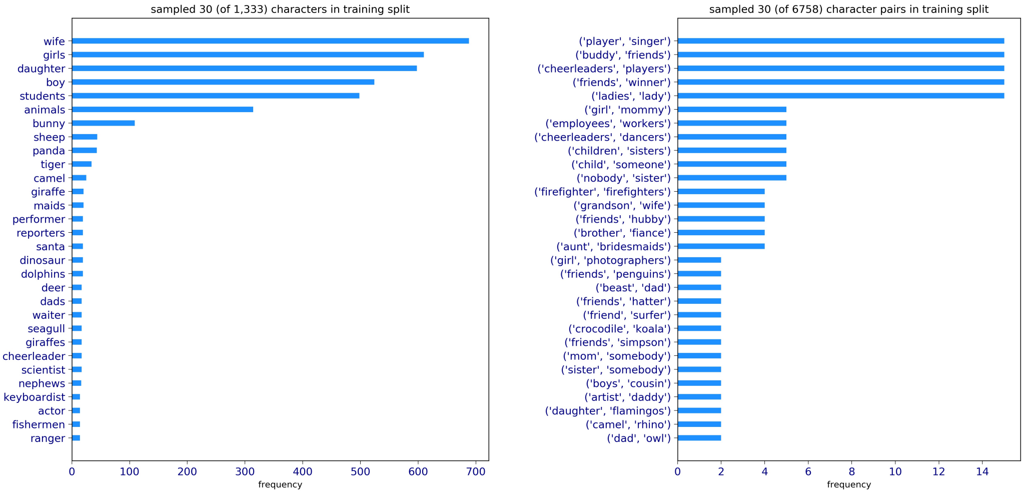


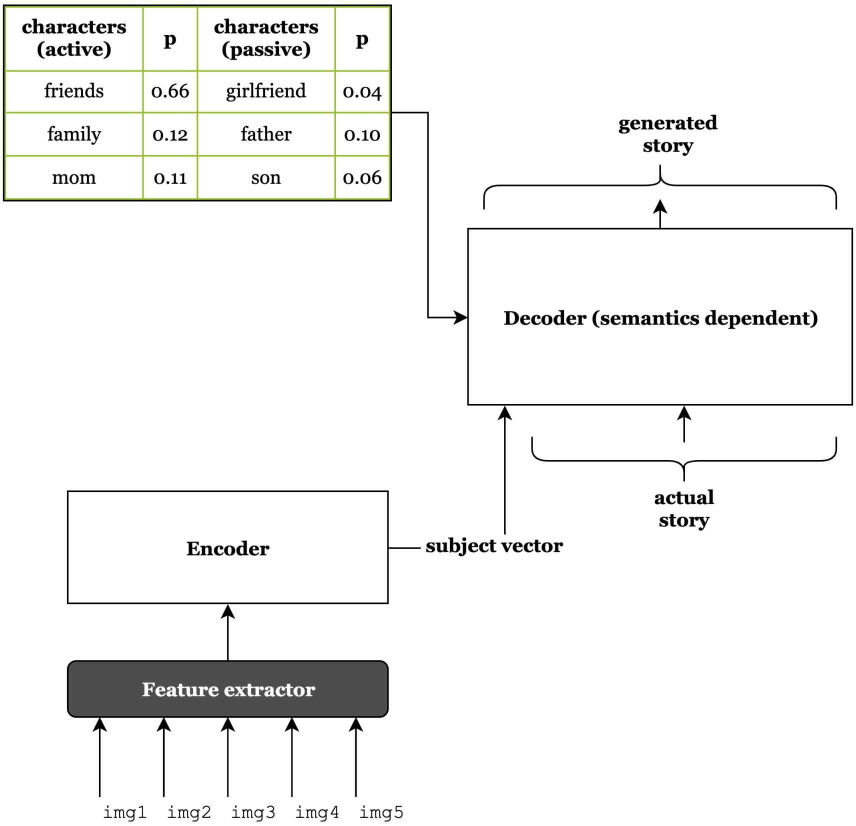
Figure 5.3: Training split character frequencies (left) and characters co-occurrence frequencies (right).

Figure 5.3 plots 30 sampled characters and co-occurring character pairs from the training split of the dataset for visualizing the skew of the frequencies. Further analysis reveals that apart from the character "friends", there is a gradual decrease in the occurrence frequencies of the other characters such as "mom" and "grandmother." Similarly, co-occurrence pairs such as ("dad", "mom"), ("friend," "friends") occur drastically more number of times than other pairs in the stories leading to an inclination bias of the story generator towards these characters, owing to the data size limitations discussed.

In the process of detecting characters, we also observed that ∼5000 distinct words failed on WordNet due to their misspellings such as ("webxites"), for being proper nouns (“cathrine”), for being an abbreviation (“geez”), and simply because they were compound words (“sing-a-long”). Though most of the models ignore these words based on a vocabulary threshold value (typically 3), language model creation without accounting for these words could adversely affect the behavior of narrative generation.

5.2 Proposed model

Our model in Figure 5.4 follows the staple encoder-decoder structure. The encoder module incorporates the image sequence features, obtained using a pre-trained convolutional network, into a subject vector. The decoder module, which is capable of considering semantics (we try several variants from [13]), uses the subject vector along with relevant character probabilities, and generates a story. Some of the variants are elaborated under Section 5.2.3.



characters (active)	p	characters (passive)	p
friends	0.66	girlfriend	0.04
family	0.12	father	0.10
mom	0.11	son	0.06

Figure 5.4: Character-centric storytelling model architecture following the encoder-decoder structure.

5.2.1 Character semantics

For semantics, we collect relevant characters concerning all data samples as a preprocessing step. We denote the characters extracted directly from the human-annotated stories of respective image-sequences as *active* characters. Using the active characters, we obtain other characters that can potentially influence the intended narrative and denote them as *passive* characters. Passive characters can be obtained using various methods. Section 5.3 describes some of these methods. The individual frequencies of these relevant characters, active and passive, are then normalized by the vocabulary size of the corpora. We identify the resulting probabilities as character semantic vectors and use them for the decoder module, as explained below.

5.2.2 Encoder

Using Resnet [18] we first obtain respective feature vectors for images in sequences. The features extracted are then provided to the encoder module, which is a standard RNN employed to learn parameters for incorporating the subjects in each of the individual feature sets into a subject vector accumulating context.

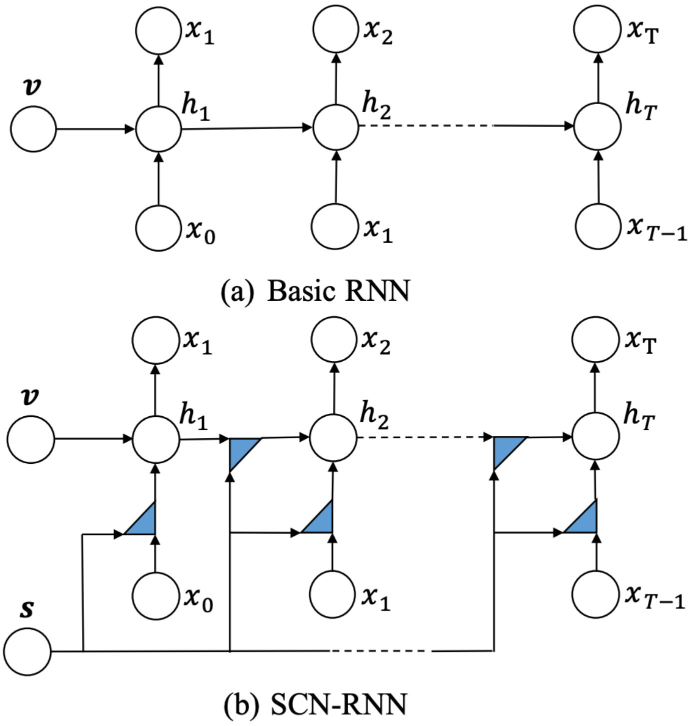
5.2.3 Decoder

Figure 5.5 presents the SCN-LSTM variant [13] of the recurrent neural network for the decoder module. The network extends each weight matrix of a conventional LSTM as an ensemble of a set of tag-dependent weight matrices, subjective to the character probabilities. Conventionally, to initialize the first timestep, we use the subject vector resulting from the encoder. The character probabilities influence the LSTM parameters utilized when decoding. Gradients propagated back to the network, nudge the parameters W towards adhering to respective character probabilities s and image sequence features v, as follows:

$$\nabla(W_{\text{gates, states}} \,|\, s, v) \;=\; \alpha \cdot \nabla_{\text{gates, states}} \quad . \tag{5.1}$$

Consequently, the encoder parameters move towards incorporating the context of the image-sequence features better. Along with the SCN-LSTM, we use two other variants of LSTM, namely LSTM-RT and LSTM-RT2 [13]. Specifically, in the network names, R denotes visual features, T denotes semantic features, and RT denotes the concatenation of both visual and semantic features.

The module is denoted as LSTM-RT if only the initial time step of the network receives the concatenated features input. Similarly, the module is denoted as LSTM-RT2 if the network receives visual Resnet features R at the first time step and the semantic features vector T at every time step of rollout. We experiment using all three semantic dependent RNN network variants for our decoder module and perform qualitative analysis.



(a) Basic RNN

(b) SCN-RNN

Figure 5.5: v and s denote the visual and semantic features respectively and each triangle symbol represents an ensemble of tag dependent weight matrices [13]

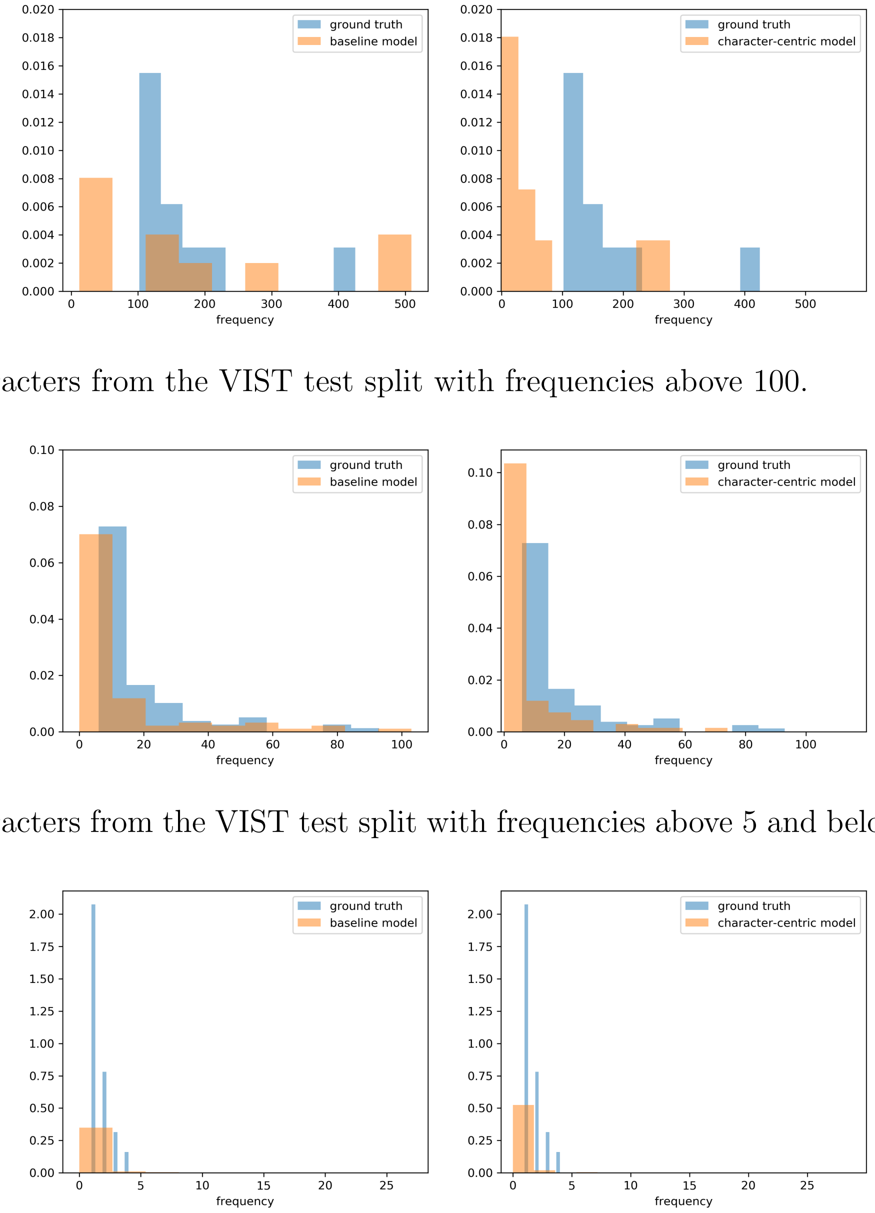
5.3 Experiments and results

The experimental setup is an extension of the baseline setting described under Section 3.1. The VIST dataset training and validation splits comprising 40071 and 4988 sentence-story pairs are utilized for learning. Before feature extraction using Resnet-152, the images are resized to be 224 dimensional and re-sampled using bi-linear interpolation. Additionally, the images are normalized with a mean of $[0.485, 0.456, 0.406]$ and a standard deviation of

$[0.229, 0.224, 0.225]$ to adhere to the ImageNet [10] collection, used for pre-training the feature extractor. Upon pre-processing the images, respective features are extracted and passed through the typical encoder module that yields a subject vector comprising the context across the image sequence. A dropout of 0.5 is applied on the extracted 2048 dimensional features and a two-layered GRU of hidden size 1000 makes up for the encoder.

As mentioned in Section 5.2.3 for the semantic decoder, experiments were performed by employing SCN, LSTM-RT, and LSTM-RT2. However, in this section, we only report details on LSTM-RT2 which worked the best for the visual storytelling task. For extracting the passive character semantics, explained in Section 5.2.1, two approaches were taken. One way was to tag all the characters co-occurring with each of the active characters as passive. The other approach was to limit the passive characters by selecting only K (hyper-parameter) most co-occurring characters. Empirical observations show that in the case of the VIST dataset, which is sparse with characters, the approach of selecting all co-occurrences as passive, worked better. The semantic feature vector of size 1470 and the sequence feature vector of size 10240 are linearly transformed to have a dimensionality of 250 for adhering to the word embedding layer dimension. At the first timestep t_0, the projections obtained for the semantic and image sequence features are concatenated as input. For the following timesteps $t_{1,\dots,T}$, respective ground truth word embeddings are concatenated with the linear projection of the semantic features and provided to the decoder GRU abiding to teacher forcing.

Other details pertaining the semantic decoder include a hidden size of 1000 for a two-layered GRU with a dropout of 0.5. Cross-entropy loss is employed as the training criterion with the Adam optimization algorithm and the learning rate of 0.0001. The model is trained for 100 epochs with a batch size of 64 and then used to generate stories for the VIST test split. The characters from the generated stories are then plotted against the true characters' distributions of the VIST test data split, to understand the influence of the trained model. Additionally, characters from the stories generated by the visual storytelling baseline model are also plotted for the purpose of comparison. All the distributions are visualized in Figure 5.6. Although the spread of characters is not obvious from the plots, the relative difference between the character-centric and baseline models is evident. Sample stories generated by the model are provided in Figure 5.7. Assessment of the model behavior and quality of the generated results are further discussed in Chapter 6.

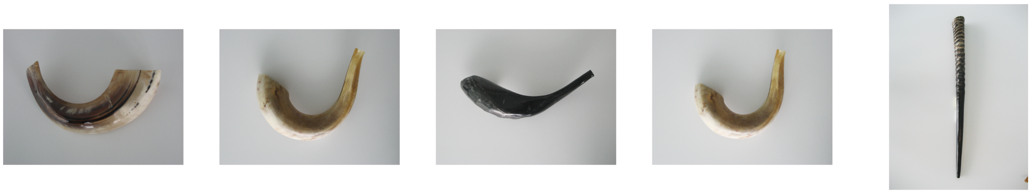


Characters from the VIST test split with frequencies above 100.

Characters from the VIST test split with frequencies above 5 and below 100.

Characters from the VIST test split with frequencies below 5.

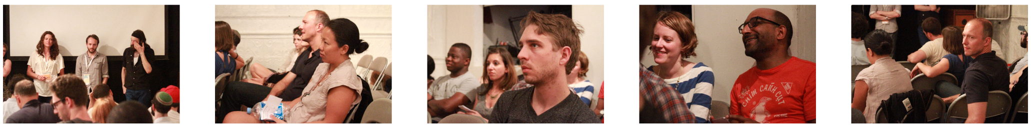
Figure 5.6: Characters' frequency distributions in the ground truth and stories generated by the trained models. Frequencies of the true 466 distinct characters from the VIST test split was found to be heavily skewed. Therefore for the purpose of analyses, the characters were segregated into three tiers, i.e., those with frequencies above 100, between 5 and 100, and below 5.



Story (Character-centric model): *i went to the museum to see the dolphins at the museum . i saw some really interesting coral . there were some amazing art installations . i also saw some penguins . i also saw some dinosaurs .*

Story (Baseline model): *i went to an art museum . there were many unique paintings . some of them were very detailed . i had a great time . i ca n't wait to go back .*

Story (human annotated): *there are many types of horns and bone . this is a ram horn and it has been shined well . next there is a horn that has been made into a pipe . a curved horn has been rubbed and shined . finally there is a straight piece of bone that has been shaped .*



Story (Character-centric model): *the crowd gathered for the awards ceremony . the speaker gave a great speech . the director gave a brief speech . the audience listened to the speaker . the cameraman was a little shy after the speech ended .*

Story (Baseline model): *i went to a meeting last night . there were a lot of people there . i had a great time with all of them . they were all very happy . i invited all of my friends .*

Story (human annotated): *everyone from the town gathered to attend the homeowner 's association meeting . [female] did n't agree with some of the proposals raised at the meeting . [male] was just thinking about going home to eat some pizza and watch tv . the presenters made a joke about pooper scoopers . [male] looked at his wife who arrived late to the meeting .*

Figure 5.7: VIST test split image sequences with stories generated by the character-centric storytelling and baseline models respectively.

Chapter 6

Discussion

Applied machine learning is about realizing the salience and validity of the theoretical concepts, from the standpoint of real-world problems. Image captioning is one such area and visual storytelling is one specific task of interest inside that area. This thesis elaborated several approaches and underlying concepts for visual storytelling. With experiments performed and results reported in the form of stories and scores, this chapter discusses the strengths, difficulties, and weaknesses by providing insights on the previous chapters. Besides, a commentary on the extended tasks that share the basic motivations and follow the same research patterns is put forth.

6.1 Closer look

Approaches for visual storytelling explained in Chapter 3 follow a very steady incremental trend. They attempt to pander to the structure and compositions of the VIST dataset while adhering to the regular captioning architectures that already proved successful. Visual storytelling formulated as a multi-class classification task inherently calls for entropy-based learning, which all the methods explained in Sections 3.1 through 3.4 followed. Entropy is without a doubt sacrosanct for understanding the behavior of the network weights, gates, and gradients. However, owing to the fact that there could never be one truly good story with respect to a visual sequence, using merely entropy as guidance, often lead to loops or dead-ends. Stories generated by the VIST baseline model in Section 3.1 are mostly sparse with both the vocabulary and maintaining consistency of the sequence context. Although the multi encoder model in Section 3.2 tries to address the aspect of context consistency by employing a previous-sentence encoder, it sometimes misguides itself into mistakes. From the sample stories provided under

Figure 3.4, the words *"two men"* could be an overstatement as a consequence of modeling the previous sentences within the story.

Both the multi decoder and GLAC models in Sections 3.3 and 3.4 shift the focus towards balancing the overall context and local relevance in each image of the sequence. Sentences generated by the five independent decoders of the multi decoder model and the cascading decoder of the GLAC model achieve the ambitious balance to an extent. The generated stories maintained tonal correctness with respect to the events in the sequences in their context. Sample story provided in Figure 3.7 with words *"excited, proud, happy"* signify the relevance of the tone. Upon empirical verification, stories generated by the GLAC model are more natural in flow without articles at the beginning of each sentence. However, stories by both the multi decoder and GLAC models are monotonous in nature, with average story length ranging from 30 to 40 words. Some of the shortcomings of these models can very well be associated with the trivial nature of the dataset.

RL based policy-gradient methods solve the restraint of the models trying to greedily fit to the ground truth story. They bring in the freedom of exploration and solve the exposure bias problem with using MLE objectives. Upon incorporating the SCST objective to optimize the GLAC model for METEOR in Section 4.1.1, the model produces a slightly better score on the VIST dataset test split. Figure 4.1 can be interpreted as a sign of the model trying to move towards yielding better METEOR scores. In the realm of this thesis, advantages of data samples at a mini-batch level are the rewards (METEOR scores) for multinomially sampled stories baselined on a single greedily inferred story. Figure 4.2 shows advantage value plots between every five epochs of training and a gradual increase in the density of samples obtaining positive advantage. However, the stories have conflicting tones and are qualitatively inferior to the regular GLAC model with cross-entropy. A comparison of story samples was provided in Figure 3.9.

The problem with vanilla policy gradient methods aiming to optimize the NLP metrics such as METEOR was explained in Section 3.5 motivating the AREL model. Alternatively, adversarial reward learning attempts to approximate a customized reward function for the VIST task. The model achieves state-of-the-art scores and generates longer stories. Experiments in Section 4.2.1 examine the behavior of AREL policy and reward models under the standard GAN objectives. Rewards for the generated and true stories by the reward model during the training, visualized in Figure 4.4, show that the generator (policy model) learns to produce stories that receive scores from the discriminator (reward model) which correlate well with the true stories. The model also achieved the highest METEOR score, summarized in Table 4.1, among all the models implemented for this thesis.

6.2 The big picture

Visual storytelling has many related sub-topics[1] such as temporal structure identification, event-episodes annotation, and character relationship modeling. Character-centric storytelling model and experiments detailed in Chapter 5 attempt to learn the underlying spread of characters and generate them in the stories with relevancy and creativity. The frequency distributions plotted in Figure 5.6 depict that the character-centric model is learning to match the true spread of characters better, compared to the regular baseline model. From among the three tiers of characters in the VIST test data split, i.e, those with frequencies above 100 (most frequent), between 5 and 100, and below 5 (least frequent), characters' distributions of the most and least frequent are moving towards the ground truth. However, for some of the sample sequences, such as the one in Figure 5.7 about *"horns and bone"*, the model exaggerates and overstates the story with characters such as *coral*, *dinosaurs* and other animals. This behavior can be due to the static character semantics vector used in the decoder module, without any fine-tuning.

Similar to the aspect of character relationships, several other facets, such as human actions, emotion modeling, and event extractions lie within the realm of visual storytelling. Related research works such as *Persona based Grounded Story Generation* [7] and *Analysis of Emotion Communication Channels in Fan-Fiction* [26] were part of the proceedings at ACL Storytelling 2019, where we also presented our work on character-centric storytelling [12]. Besides, recent developments in scene-graph tasks for representing visual information can be adapted for the visual storytelling task by extracting and learning regions within image sequences with relationships.

[1] http://www.visionandlanguage.net/workshop2019/cfp.html

Chapter 7

Conclusions

This thesis discussed the visual storytelling task to its fullest current extent. Upon introducing the problem formally and providing the motivations behind its conception, an extensive background coverage of respective parent and sibling domains pertaining captioning was reported. Captioning as a topic is interesting considering its uniqueness in amalgamating both visual and textual modalities. With this motivation, details about relevant methods from the literature were outlined. Analyses were performed on the VIST dataset, a publicly available source for the visual storytelling task. The present state and challenges with automatically evaluating and scoring the texts generated by machine learning models were discussed.

Existing architectures which were proposed for the visual storytelling task were discussed in detail with relevant motivations and objectives in perspective. Results were presented in the form of both image sequence-story pairs and NLP scores. Along with replicating and re-implementing the existing designs, remodeling by utilizing distinctive learning mechanisms was carried out. Behavior of policy-gradient methods, which are gaining traction in every domain of deep learning, were examined from the visual storytelling angle. Also, the realm of adversarial learning models were scrutinized and compared against each other qualitatively.

Additionally, the sub-domain of character relationships within the topic of visual storytelling was investigated. Character semantics of the VIST dataset were extracted and broadly analyzed. A design for modeling the extracted semantic features along with the standard visual features was proposed. Learning behaviors, inference stage performances, and subjective aspects of the generated results were comprehensively discussed.

Means for analyzing the actual influence of visual and textual modalities on the model outcomes are essential for a thorough understanding of the paradigm and is one of the directions in which this work can be taken forward. Granular experimentation with region-level features of the visual data and understanding of sentence-level rewards within stories can potentially improve the results. Reverse engineering a customized scorer from models which learn through reinforcement, for automatically assessing the generated stories is another facet to research forward.

Bibliography

[1] Aafaq, N., Gilani, S. Z., Liu, W., and Mian, A. Video description: A survey of methods, datasets and evaluation metrics. *CoRR abs/1806.00186* (2018).

[2] Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., Courville, A. C., and Bengio, Y. An actor-critic algorithm for sequence prediction. *CoRR abs/1607.07086* (2016).

[3] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *ArXiv 1409* (09 2014).

[4] Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S. J., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., Schmidt, L., Shangguan, J., Siskind, J. M., Waggoner, J. W., Wang, S., Wei, J., Yin, Y., and Zhang, Z. Video in sentences out. *CoRR abs/1204.2742* (2012).

[5] Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR abs/1506.03099* (2015).

[6] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. A neural probabilistic language model. *J. Mach. Learn. Res. 3* (Mar. 2003), 1137–1155.

[7] Chandu, K., Prabhumoye, S., Salakhutdinov, R., and Black, A. W. “my way of telling a story”: Persona based grounded story generation. In *Proceedings of the Second Workshop on Storytelling* (Florence, Italy, Aug. 2019), Association for Computational Linguistics, pp. 11–21.

[8] Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR abs/1406.1078* (2014).

[9] Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR abs/1406.1078* (2014).

[10] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09* (2009).

[11] Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., and Mitchell, M. Language models for image captioning: The quirks and what works. *CoRR abs/1505.01809* (2015).

[12] Ferraro, F., Huang, T.-H. K., Lukin, S. M., and Mitchell, M., Eds. *Proceedings of the Second Workshop on Storytelling* (Florence, Italy, Aug. 2019), Association for Computational Linguistics.

[13] Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., Carin, L., and Deng, L. Semantic compositional networks for visual captioning. *CoRR abs/1611.08002* (2016).

[14] Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR abs/1311.2524* (2013).

[15] Gonzalez-Rico, D., and Pineda, G. F. Contextualize, show and tell: A neural visual storyteller. *CoRR abs/1806.00738* (2018).

[16] Graff, D., Kong, J., Chen, K., and Maeda, K. English gigaword. *Linguistic Data Consortium, Philadelphia 4*, 1 (2003), 34.

[17] Graves, A. Generating sequences with recurrent neural networks. *CoRR abs/1308.0850* (2013).

[18] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR abs/1512.03385* (2015).

[19] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Comput. 9*, 8 (Nov. 1997), 1735–1780.

[20] Hodosh, M., Young, P., and Hockenmaier, J. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res. 47*, 1 (May 2013), 853–899.

[21] Huang, Q., Gan, Z., Çelikyilmaz, A., Wu, D. O., Wang, J., and He, X. Hierarchically structured reinforcement learning for topically coherent visual story generation. *CoRR abs/1805.08191* (2018).

[22] Huang, T. K., Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R. B., He, X., Kohli, P., Batra, D., Zitnick, C. L., Parikh, D., Vanderwende, L., Galley, M., and Mitchell, M. Visual storytelling. *CoRR abs/1604.03968* (2016).

[23] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37* (2015), ICML'15, JMLR.org, pp. 448–456.

[24] Johnson, J., Karpathy, A., and Li, F. Densecap: Fully convolutional localization networks for dense captioning. *CoRR abs/1511.07571* (2015).

[25] Karpathy, A., and Li, F. Deep visual-semantic alignments for generating image descriptions. *CoRR abs/1412.2306* (2014).

[26] Kim, E., and Klinger, R. An analysis of emotion communication channels in fan fiction: Towards emotional storytelling. *CoRR abs/1906.02402* (2019).

[27] Kim, T., Heo, M., Son, S., Park, K., and Zhang, B. GLAC net: Glocal attention cascading networks for multi-image cued story generation. *CoRR abs/1805.10973* (2018).

[28] Kiros, R., Salakhutdinov, R., and Zemel, R. Multimodal neural language models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32* (2014), ICML'14, JMLR.org, pp. II–595–II–603.

[29] Kiros, R., Salakhutdinov, R., and Zemel, R. S. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR abs/1411.2539* (2014).

[30] Krause, J., Johnson, J., Krishna, R., and Fei-Fei, L. A hierarchical approach for generating descriptive image paragraphs. *CoRR abs/1611.06607* (2016).

[31] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D. A., Bernstein, M. S., and Li, F. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR abs/1602.07332* (2016).

[32] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (USA, 2012), NIPS'12, Curran Associates Inc., pp. 1097–1105.

[33] Lamb, A., Goyal, A., Zhang, Y., Zhang, S., Courville, A., and Bengio, Y. Professor forcing: A new algorithm for training recurrent networks, 2016.

[34] Lavie, A., and Agarwal, A. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation* (Stroudsburg, PA, USA, 2007), StatMT '07, Association for Computational Linguistics, pp. 228–231.

[35] Li, S., Tao, Z., Li, K., and Fu, Y. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence PP* (01 2019), 1–16.

[36] Liang, X., Hu, Z., Zhang, H., Gan, C., and Xing, E. P. Recurrent topic-transition GAN for visual paragraph generation. *CoRR abs/1703.07022* (2017).

[37] Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR abs/1405.0312* (2014).

[38] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations* (2014), pp. 55–60.

[39] Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology* (Stroudsburg,

PA, USA, 1994), HLT '94, Association for Computational Linguistics, pp. 114–119.

[40] Mikolov, T., Kopecky, J., Burget, L., Glembek, O., and ?Cernocky, J. Neural network based language models for highly inflective languages. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (Washington, DC, USA, 2009), ICASSP '09, IEEE Computer Society, pp. 4725–4728.

[41] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (USA, 2013), NIPS'13, Curran Associates Inc., pp. 3111–3119.

[42] Miller, G. A. Wordnet: A lexical database for english. *Commun. ACM 38*, 11 (Nov. 1995), 39–41.

[43] Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., and Daumé, III, H. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2012), EACL '12, Association for Computational Linguistics, pp. 747–756.

[44] Mnih, A., and Hinton, G. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning* (New York, NY, USA, 2007), ICML '07, ACM, pp. 641–648.

[45] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Stroudsburg, PA, USA, 2002), ACL '02, Association for Computational Linguistics, pp. 311–318.

[46] Park, C. C., and Kim, G. Expressing an image stream with a sequence of natural sentences. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Cambridge, MA, USA, 2015), NIPS'15, MIT Press, pp. 73–81.

[47] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W* (2017).

[48] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *In EMNLP* (2014).

[49] Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks, 2015.

[50] Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR abs/1506.01497* (2015).

[51] Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. Self-critical sequence training for image captioning. *CoRR abs/1612.00563* (2016).

[52] Senina, A., Rohrbach, M., Qiu, W., Friedrich, A., Amin, S., Andriluka, M., Pinkal, M., and Schiele, B. Coherent multi-sentence video description with variable level of detail. *CoRR abs/1403.6173* (2014).

[53] Seog Han, S., Hun Park, G., Lim, W., Shin Kim, M., Na, J.-I., Park, I., and Eun Chang, S. Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network. *PLOS ONE 13* (01 2018), e0191493.

[54] Shen, X. A survey of object classification and detection based on 2d/3d data. *CoRR abs/1905.12683* (2019).

[55] Shetty, R., Tavakoli, H. R., and Laaksonen, J. Image and video captioning with augmented neural architectures. *IEEE MultiMedia 25*, 2 (2018), 34–46.

[56] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014).

[57] Smilevski, M., Lalkovski, I., and Madjarov, G. Stories for images-in-sequence by using visual and narrative components. *CoRR abs/1805.05622* (2018).

[58] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 1999), NIPS'99, MIT Press, pp. 1057–1063.

[59] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *CoRR abs/1512.00567* (2015).

[60] Vedantam, R., Zitnick, C. L., and Parikh, D. Cider: Consensus-based image description evaluation. *CoRR abs/1411.5726* (2014).

[61] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. *CoRR abs/1411.4555* (2014).

[62] Wang, X., Chen, W., Wang, Y., and Wang, W. Y. No metrics are perfect: Adversarial reward learning for visual storytelling. *CoRR abs/1804.09160* (2018).

[63] Williams, R. J., and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation 1*, 2 (June 1989), 270–280.

[64] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. *CoRR abs/1502.03044* (2015).

[65] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., and Courville, A. Video description generation incorporating spatio-temporal features and a soft-attention mechanism. *arXiv preprint arXiv:1502.08029* (2015).

[66] Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR abs/1609.05473* (2016).