

Towards an Improved Model for Visual Storytelling

by

Yatri Modi

Bachelor of Engineering, University of Mumbai, Mumbai, India, 2018

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2020

Chicago, Illinois

Defense Committee:

Prof. Natalie Parde, Chair and Advisor

Prof. Barbara Di Eugenio

Prof. Sathya Ravi

ACKNOWLEDGMENTS

My sincere thanks to Dr. Natalie Parde without whom this research would just not have been possible. Her guidance and constant support gave me the confidence to complete this work. Thank you for believing in me and giving me this opportunity.

I am also very thankful to all the professors at UIC who have taught me because without the indispensable knowledge they gave me, I just would not have been able to complete my research.

I am very grateful to my parents, Manoj and Hemali Modi, for consistently motivating me and supporting me throughout this journey. Your teachings and values have made it possible for me to be writing down this thesis today. Thank you for helping me achieve this dream.

My sister, Maitri, my source of happiness and joy, thank you for always checking on me and uplifting my spirits. My dear roommates, Shivani and Tanvi, you both made Chicago feel like home. Chinmay and Utsav, UIC wouldn't have been the same without you both!

Special thanks to my classmates and friends from my undergrad, BE B, for preparing me to face the real world.

YM

CONTRIBUTION OF AUTHORS

This study contains work from Modi and Parde,2019 [1]. I am the first author of the work and my advisor, Prof Natalie Parde, is the second author and has co-written/ edited it with me.

TABLE OF CONTENTS

<u>CHAPTER</u>		<u>PAGE</u>
1	INTRODUCTION	1
1.1	What is Visual Storytelling?	1
1.2	Visual Storytelling Challenge 2018	3
1.3	Outline	4
2	RELATED WORK	5
3	DATA	9
3.1	Visual Storytelling Dataset	9
4	AN ANALYSIS OF ERRORS IN EXISTING VISUAL STORY-TELLING MODELS	11
4.1	Overview	11
4.2	Model Summaries	12
4.2.1	Adversarial Reward Learning	12
4.2.2	GLocal Attention Cascading Networks	13
4.2.3	Contextualize, Show and Tell	14
4.3	Experimental Setup	15
4.4	Evaluation Metrics	16
4.5	Results	16
4.5.1	Error Categories	17
4.5.2	Discussion	24
4.6	Recommendations	26
5	THE PROPOSED MODEL	29
5.1	Methods	29
5.2	Natural Language Generation	29
5.3	Artificial Neural Networks	30
5.4	Recurrent Neural Networks	31
5.5	Long Short Term Memory	31
5.6	Convolutional Neural Networks	35
5.7	Encoder-Decoder Networks	37
5.8	Hierarchical Context-Based Network	38
5.8.1	Hierarchical Encoder Network	38
5.8.1.1	Image Sequence Encoder	38
5.8.1.2	Image Description Encoder	38
5.8.2	Sentence Decoder	39

TABLE OF CONTENTS (Continued)

<u>CHAPTER</u>		<u>PAGE</u>
	5.9 Our Modification to HCBNet	40
	5.9.1 Coattention for Visual Storytelling	41
6	EXPERIMENTAL SETUP	46
7	EVALUATION AND RESULTS	47
	7.1 Quantitative Evaluation	47
	7.1.1 BERTScore	47
	7.2 Results and Discussion	48
	7.3 Qualitative Evaluation	50
8	CONCLUSION AND FUTURE WORK	56
	APPENDIX	57
	CITED LITERATURE	60
	VITA	65

LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
I	Example stories associated with each error category. This table is taken from Modi and Parde, 2019 [1].	20
II	Example stories associated with each error category. (TABLE I CONTINUED..)	21
III	Example stories associated with each error category. (TABLE II CONTINUED)..	22
IV	Example scoring anomalies, including the anomalous scores for each story.	23
V	Frequency (in terms of overall percentage) of the most common error types across all 1010 generated test stories by AREL and GLAC-Net and 1938 generated test stories by Contextualize, Show and Tell. This table is taken from Modi and Parde, 2019 [1].	24
VI	Performance as reported in the source papers [2, 3]. BLEU-RL, METEOR-RL, and CIDEr-RL were baseline reinforcement learning approaches using BLEU, METEOR, and CIDEr scores as their reward functions, respectively [2].	49
VII	BERTScore Precision (P_{BERT}), Recall(R_{BERT}) and F1 Score $F1_{BERT}$. The maximum values are made bold.	50
VIII	Experimental results showing some word properties	53

LIST OF FIGURES

<u>FIGURE</u>		<u>PAGE</u>
1	A sequence of images from the VIST dataset. The image is taken from Modi and Parde, 2019 [1]	2
2	RNN with its unrolled version through time	32
3	LSTM cell with update, input and output gates	33
4	Sequence Processing in a Bidirectional RNN	34
5	An example of a Convolutional Neural Network with convolution, pooling and fully connected (FC) layers.	36
6	Coattention for Visual Storytelling. The affinity matrix L has been omitted but the attention weights A^D, A^V have been shown directly. W is the word embedding of the previously generated word by the SD. . .	43
7	The Proposed Model with Coattention added to HCBNet (Nahian et al. 2019 [4]).	44
8	Histograms showing the count of unique 1-grams, 2-rams, 3-grams and 4-grams in the various models.	51
9	The different visual stories generated by the baselines and our model.	54
10	A very imaginative and good quality visual story generated by our model.	55
11	A very imaginative and good quality visual story generated by our model.	59

LIST OF ABBREVIATIONS

VIST	Visual Storytelling
DII	Descriptions of Images in Isolation
SIS	Story in Sequence
UIC	University of Illinois at Chicago
AREL	Adversarial Reward Learning
GLACNet	GLocal Attention Cascading Networks
CST	Contextualize, Show and Tell
METEOR	Metric for Evaluation of Translation with Explicit ORdering
BLEU	the Bilingual Evaluation Understudy
ROUGE-L	Recall-Oriented Understudy for Gisting Evalua- tion - Longest Common Subsequence
CIDEr	Consensus- based Image Description Evaluation
ANN	Artificial Neural Network
DNN	Deep Neural Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network

LIST OF ABBREVIATIONS (Continued)

LSTM	Long Short Term Memory
BiLSTM	Bidirectional Long Short Term Memory
HCBNet	Hierarchical Context Based Network
ISE	Image Sequence Encoder
IDE	Image Description Encoder
DE	Description Encoder
IE	Image Encoder
SD	Sentence Decoder
BERT	Bidirectional Encoder Representations from Trans- formers

SUMMARY

Visual storytelling is an intriguing and complex task that only recently entered the language and vision research arena. The task focuses on generating human-like, coherent and visually grounded stories from a sequence of images while maintaining the context over these images. In this study I survey recent advances in the field and conduct a thorough error analysis of three approaches to visual storytelling. I categorize and provide examples of common types of errors, and identify key shortcomings in prior work. Later, I make recommendations for addressing these limitations, and propose an improved model for visual storytelling: a hierarchical encoder-decoder network, with co-attention over the images and their natural language literal descriptions. I assess the performance of this model at generating visual stories. Finally, I experiment with a novel metric, BertScore [5], as an alternative to human evaluation.

CHAPTER 1

INTRODUCTION

Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

1.1 What is Visual Storytelling?

Artificial intelligence continues to evolve, making it increasingly plausible to develop models that interpret vision and language in a humanlike manner. A crucial element of such models is the capacity to not only match images with surface-level descriptions, but to infer deeper contextual meaning. Recent literature has begun to refer to this task as *visual storytelling*: the generation of a cohesive, sequential set of natural-language descriptions across multiple images [6]. Visual storytelling is distinct from image captioning in that the text generated is oftentimes subjective, hinges on contextual image order, and typically employs more abstract and dynamic terms. We illustrate the dichotomy between the two more concretely in terms of possible sets of sentences¹ for the images in Figure 1 on the next page.

¹Real samples (with punctuation and capitalization edited in some cases to increase readability) from the VIST dataset: <http://visionandlanguage.net/VIST/dataset.html>



Figure 1. A sequence of images from the VIST dataset. The image is taken from Modi and Parde, 2019 [1]

Sentence Set 1: (1) A woman looking at a collection of tribal masks on the wall. (2) Three skulls of varying sizes ordered from largest to smallest. (3) A top view of a book about mythical creatures. (4) Three people standing in a store looking at the products. (5) An old traveling wagon that is on display.

Sentence Set 2: (1) I went to the natural history museum today. (2) Their evolution display was very interesting. (3) They had an area for cryptozoology. (4) They also have a gift shop. (5) My favorite was this real covered wagon from 200 years ago.

The first is a set of traditional image captions, whereas the latter represents a visual story. Note that the former presents factual descriptions of the images in isolation from one another. The latter also describes the images, but places stronger emphasis on the development of a cohesive narrative underlying the image sequence.

High-performing visual storytelling approaches will enable growth for a variety of applications, many of which are associated with language understanding tasks. They may also hold promise as a tool for assistive technology. For instance, it is relatively common for users to upload large photo albums to social media platforms without including any image descriptions at all, making these albums inaccessible to those with sight impairments. Visual storytelling could bridge this gap by automatically generating descriptive narratives for these albums.

1.2 Visual Storytelling Challenge 2018

Most of the initial work towards visual storytelling was conducted in the context of the 2018 Visual Storytelling Challenge; thus, we focus our analysis on methods employed by the participating teams. The challenge required participants to make AI systems capable of generating human-like stories from a sequence of images as input. It had (1) an *Internal Track* that constrained participants such that they could train only on data from the Visual Storytelling (VIST) Dataset, described further in Section 4.1, and use pretraining data only from any version of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)¹ and any version of the

¹A well-known annual competition that challenges researchers to solve a variety of large-scale object and image detection tasks [7]: <http://image-net.org/challenges/LSVRC/>.

Penn Treebank;¹ and (2) an *External Track* that allowed participants free reign when training, with the only requirement being that all training data be made publicly accessible if it was not already. The challenge evaluated the quality of the generated stories using both an automatic metric (METEOR [9] , described in further detail in 5.4) and human ratings corresponding to the following characteristics: (1) focus, (2) structure and coherence, (3) inclination to share, (4) likelihood of being written by a human, (5) visual grounding quality, and (6) level of detail.² The winning team for the challenge was DG-DLMX [10]. Other participating teams included UCSB-NLP [2], SnuBiVtt [3], and NLPSA501 [11].

1.3 Outline

Chapter 1 introduces visual storytelling and the shared task that took place for it. Chapter 2 discusses related work.

The dataset is described in detail in Chapter 3 followed by the error analysis of visual storytelling approaches in Chapter 4. We discuss the details of our proposed model in Chapter 5 followed by the experimental setup used for the same in Chapter 6.

The evaluation and the results we obtained by conducting experiments with the proposed model are mentioned in Chapter 7. Finally, we end the discussion with our conclusion and future work in Chapter 8.

¹ [8]: <https://catalog.ldc.upenn.edu/LDC99T42>.

²Human judgements were solicited using Amazon Mechanical Turk (<https://www.mturk.com/>).

CHAPTER 2

RELATED WORK

(Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.)

A small but growing body of research has investigated visual storytelling. We perform an in-depth error analysis of the work done by UCSB-NLP [2], SnuBiVtt [3], and DG-DLMX [10] for the Visual Storytelling Challenge. These are the three teams who have released publicly available source code to date. We describe these models in further detail in Chapter 5. The other team participating in the challenge was NLPSA501 [11]. NLPSA501 introduced a convolutional neural network (CNN) and gated recurrent unit (GRU) encoder-decoder model that incorporated an inter-sentence diverse beam search as a way to reduce redundancy in the generated stories. We could not analyze their model’s output as we did for those by UCSB-NLP, DG-DLMX and SnuBiVtt, due to the lack of available implementations or generated stories.

Outside of the Visual Storytelling Challenge, several other groups have also explored the task. Huang et al. [6] published the original paper introducing the visual storytelling task, highlighting the differences between storytelling and image captioning. The authors used GRUs for both encoding the image and decoding the story. Lukin et al [12] defined a pipeline for visual storytelling consisting of Object Detection, Single-Image Inferencing, and Multi-Image Narra-

tion steps. Yu et al. [13] employed an alternate pipeline comprised of Album Encoder, Photo Selector, and Story Generator stages. Agrawal et al.’s [14] approach focuses on identifying proper sequences for existing story sentences, rather than on generating those sentences themselves. Jain et al. [15] explored a phrase-based and syntax-based statistical machine translation approach as a vehicle for story generation using text but no images from the VIST dataset. The approaches developed for the Visual Storytelling Challenge were designed to be improvements upon Huang et al.’s [6] model.

The task of visual storytelling is still in its infancy, and to date there exists no comprehensive review of prior work in this area. Our analysis fills this void, by summarizing relevant work in a shared context and providing concrete comparisons and example output when possible. This in turn allows us to identify core areas for improvement in our implementation, recommending specific actions to address these current limitations. Our hope is that the analysis can also serve as a useful launchpad for us and other researchers aspiring to work in the visual storytelling domain.

Since publishing our original analysis [1], additional work has also been published in the visual storytelling domain. We build upon some of this work in our proposed model. Specifically, we use the Hierarchical Context-based Network (HCBNet) architecture proposed by Nahian et al. [4] as the base for our network. HCBNet is a hierarchical, context-based neural network. Its name is derived from the fact that the network makes hierarchical use of natural language descriptions of images along with the images themselves to generate the visual stories. Nahian et al. claim that this allows them to learn the expected sequence of events that take place from

the input images and produce a visually grounded and cohesive story. We discuss this model in more detail in Chapter 6.

Recent work by Yang et al. [16], Wang et al. [17] and Hsu et al. [18] has sought to leverage supplemental context, making use of knowledge graphs for visual storytelling. Specifically, all three works make use of Faster R-CNNs pretrained with the Visual Genome Dataset [19]. Yang et al. [16] first distill the terms from images using an image-to-term model, then enrich the word set using scene graphs and produce term paths. Finally, they use the Transformer architecture to generate the stories. They make use of external data for all three of those tasks, outperforming GLACNet [3] in a human evaluation.

Hsu et al. 2019a [18] incorporate knowledge graphs generated using external data to extract commonsense knowledge. They consider the key concepts of images as nodes in a graph, which are surrounded by nodes representing imaginary/commonsense concepts. They extract the commonsense concepts and combine them with semantic visual features, attempting to maximize the semantic similarity in the output during optimization. The CIDEr score [20] for this model outperforms other state-of-the-art models.

Wang et al. [17] propose graph convolutional networks for extracting local, fine grained scene graphs from individual images and then finding the cross-relations between them using temporal convolutional networks. Visual attention is used in the decoder with the knowledge graph to generate the stories. This model is more informative and outperforms AREL [2] in a human evaluation.

Finally, recent work that does not incorporate knowledge graphs includes that of Jung et al. [21] and Hu et al. [22]. Jung et al. [21] introduced a “Hide and Tell” network. In this network, some images are hidden from the input stream using curriculum learning, basically the number of images hidden from a single training sequence increases with time and starts with showing all images to the model first, and the model tries to fill in the visual gaps by imagining what non-local relations could have been between the missing and shown images. The model outperforms other alternatives in both human and automatic evaluations. Hu et al. [22] instead optimize their model, ReCO-RL, using three human evaluation criteria (relevance, coherence, and expressiveness) to generate high quality stories. Specifically, they employ a reinforcement learning framework with three reward functions to score the above mentioned criteria. Their model outperforms AREL [2] by a wide margin.

CHAPTER 3

DATA

Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

3.1 Visual Storytelling Dataset

Most visual storytelling work to date has been trained and evaluated using the VIST Dataset [6]. VIST is the first publicly available dataset for sequential vision-to-language tasks, and consists of sequences or “albums” of images wherein each image is paired with two types of captions; namely, descriptions of images in isolation (DII), and stories of images in sequence (SIS). The images were originally downloaded from Flickr (<https://www.flickr.com/>). In total, the dataset comprises 10,117 Flickr albums containing 210,819 unique photos.

Amazon Mechanical Turk (AMT) workers selected subsets of five images per album about which to write sequential, cohesive stories. The dataset contains 50,200 story sequences overall; these are divided into subsets of 40,155 training, 4,990 validation and 5,055 testing stories. Five written stories were collected per album. Three standalone descriptions per image (DII, first defined above) were also collected separately using the image captioning interface used to build the COCO image caption dataset [23].

In both the stories and descriptions, all people names were replaced with generic MALE/FEMALE tokens, and all named entities were replaced with their entity type (e.g., LOCATION). A small number of broken images were filtered from VIST by most research groups. For concrete examples of DII and SIS from VIST, we refer readers to Figure 1, where Sentence Sets 1 and 2 (see Chapter 1) are from the DII and SIS subsets, respectively.

When developing our own model, we selected only those stories which had DII available for every image, following the practice that Nahian et al. [4] took when training HCBNet. This resulted in a total of 26,905 training stories, 3,354 validation stories, and 3,385 test stories.

CHAPTER 4

AN ANALYSIS OF ERRORS IN EXISTING VISUAL STORYTELLING MODELS

Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

4.1 Overview

In this chapter, we conduct an analysis of three of the approaches submitted to the Visual Storytelling Challenge: AREL [2], GLACNet [3], and Contextualize, Show and Tell [10]. We selected these approaches as the focus of our work for two reasons. First, all were publicly available and well-documented, ensuring easy replicability. Other existing visual storytelling models would have required reimplementing or a lot of processing power. Doing so introduces the possibility of unintentionally crippling performance (e.g., when setting required but unreported parameters), which we wished to avoid. Second, all were very recent models, representing some of the current state of the art in visual storytelling. We summarize AREL, GLACNet, and Contextualize, Show and Tell in Section 5.1, and refer readers to the original papers for fuller detail.

4.2 Model Summaries

4.2.1 Adversarial Reward Learning

Adversarial Reward Learning (AREL) [2] is an adversarial reinforcement learning approach that makes use of two models: a policy model, followed by a reward model. The policy model is an encoder-decoder model utilizing a CNN-recurrent neural network (RNN) architecture, used to generate new stories. Specifically, a pre-trained CNN is fed a sequence of 5 images as input to extract high-level image features which are further encoded as visual context vectors using bidirectional GRUs. The outputs of the encoder are then fed into a GRU-RNN decoder to generate sub-stories for the image sequence in parallel. The sub-stories are concatenated to form a single full story. The CNN-based reward model is applied to every sub-story to compute its partial reward, and from the input sequence embeddings, n-gram features are extracted using convolution kernels of different sizes and passed through pooling layers. Image features are concatenated with these sentence representations and passed through a fully connected layer to obtain the final reward. The objective of the story generation policy was to maximize the similarity between a Reward Boltzmann distribution and itself. The first model optimized the policy to minimize the KL divergence [24] between itself and the Boltzmann Distribution, and the second model attempted to (a) minimize the KL divergence with the empirical distribution, and (b) maximize the KL divergence with the approximated policy distribution, with the objective of distinguishing between human and machine generated stories.

AREL outperformed a generative adversarial network (GAN) model, a cross-entropy model, and other baselines and achieved state-of-the-art results across both automated and human

metrics. The human metrics considered included both a Turing test (in which annotators attempted to guess which of two stories was written by a human) and pairwise comparisons measuring relevance, expressiveness, and concreteness.

4.2.2 GLocal Attention Cascading Networks

GLocal Attention Cascading Networks (GLACNet) [3] also uses an encoder-decoder architecture, but it adds a hard attention mechanism which stresses feeding both the local image features and the overall context to the decoder as input. The image-specific features are extracted using a 152-layer residual network [25]. Those features are fed sequentially into a bidirectional LSTM, which then produces the global context vectors. The global context and local image features are combined to form *glocal* vectors and passed through fully connected layers. The output is concatenated with word tokens and fed to the decoder (LSTM) as input. Thus, five glocal vectors for each image are fed into the decoder one after another, creating a cascading mechanism by passing the hidden state of one sentence generator as the initial hidden state of the next sentence generator.

To validate that all components of the GLACNet architecture contributed to the model’s performance, Kim et al. [3] conducted an ablation study in which the cascading, global attention, local attention, and post-processing routines were removed one at a time, comparing perplexity and METEOR [9] scores between conditions as well as with a standalone LSTM sequence-to-sequence (Seq2Seq) model and the full GLACNet model. The full GLACNet model exhibited the best performance on the hidden test set in the Visual Storytelling Challenge (however, it

was not declared the winner since the authors submitted their results late). HCBNet [4], the basis of our proposed model, is built on top of GLACNet.

4.2.3 Contextualize, Show and Tell

Contextualize, Show and Tell [10] won the 2018 Visual Storytelling Challenge. The model uses an encoder LSTM to read in the image representations one by one for every image in a sequence. The image representations are generated using Inception V3 [26]. Five decoders, again LSTMs, then read in the image embedding as input. The first hidden state of each decoder is initialized using the last hidden state of the encoder to provide the model with global context. Gonzalez-Rico and Pineda [10] obtained the final story by concatenating the outputs of the model’s five decoders.

As part of the Visual Storytelling Challenge, the model was evaluated on public and hidden test sets using both human evaluation and an automated metric (METEOR). METEOR scores of 30.88 and 31 were obtained on the public and hidden test sets, respectively.¹ Human evaluation scores were collected via Amazon Mechanical Turk.² Crowd workers evaluated six aspects of each story using a Likert scale. Each worker was asked to indicate the degree to which: 1) the story was focused, 2) the story had good structure and coherence, 3) the worker would share the story, 4) the worker thought the story was written by a human, 5) the story was visually grounded, and 6) the story was detailed. In summing the average scores received for each

¹Gonzalez-Rico and Pineda [10] reported a METEOR score of 34.4 on the standard VIST test set.

²<https://www.mturk.com/>

criterion, Contextualize, Show and Tell achieved a score of 18.498, whereas human-generated stories achieved a score of 23.596.

4.3 Experimental Setup

We trained and evaluated AREL according to the instructions provided in its publicly available Github repository.¹ However, we modified the source code slightly such that we were able to obtain the individual METEOR scores for each predicted story in the test set. This helped us in performing an in-depth error analysis of the generated stories and determining how well the automatic metrics were at scoring the stories. Training the model took around 2 weeks on a 3.5 GHz Intel Core i5 CPU with 16 GB RAM.²

The GLACNet code is also publicly available.³ We trained and evaluated the model using an NVIDIA Tesla P100 GPU instance on Google Cloud Platform. The model took one week to finish training. The original source code only provided an average METEOR score across all generated stories after testing. Thus, we added code to produce the METEOR score for each story. We will make all adapted source code publicly available online to ensure easy replicability.

¹<https://github.com/littlekobe/AREL>

²Extenuating circumstances limited our hardware resources in the midst of our AREL evaluation. Training would have undoubtedly been quicker using GPUs, as was done in the original paper [2].

³<https://github.com/tkim-snu/GLACNet>

The source code for Contextualize, Show and Tell is available online as well.¹ The authors personally sent us the generated stories, so we did not re-implement their model. We have directly included their METEOR results in our evaluation.

4.4 Evaluation Metrics

Common metrics for evaluating visual storytelling models include METEOR [9], BLEU [27], CIDEr [20], and ROUGE-L [28]. METEOR, the primary metric considered in the Visual Storytelling Challenge, calculates the alignment between the machine-generated hypotheses and the reference stories based on the exact, stem, synonym, and paraphrase matches between words and phrases. CIDEr is a consensus based metric that is supposed to capture human judgement well. ROUGE-L calculates F-score and not just recall. It is based on the longest common subsequence. BLEU does not take into account recall and only considers n-gram precision. It examines overlap between the output and reference translations, assigning a penalty to shorter outputs. While AREL was evaluated using METEOR as well as the other metrics, GLACNet was evaluated using only METEOR scores and measures of perplexity. Contextualize, Show and Tell was also evaluated using only METEOR. We generated scores for the remaining metrics as well for GLACNet and Contextualize, Show and Tell to aid our analysis.

4.5 Results

We defined a threshold METEOR score of 25, with stories scoring below this threshold considered as serious errors. This threshold was chosen following a manual assessment of the

¹<https://github.com/dgonzalez-ri/neural-visual-storyteller>

predicted stories, with METEOR < 25 representing a medium at which there existed both a sizable number of errors, and a sample of generated stories that were of noticeably low quality. Stories having a METEOR score ≥ 50 were also analyzed for any anomalies (e.g., bad stories with high scores).

Some metrics (CIDEr and BLEU-4) produced scores of 0 for many stories in both models. Upon manual analysis we found many of these stories to be sensible. Other work has confirmed that BLEU-3 and CIDEr scores do not correlate well with human evaluations [2].

We systematically analyzed errors in each story and made notes indicating characteristics contributing to the error (including those that rendered the predicted stories to be completely meaningless or incoherent). In the process, we also identified mechanisms by which those errors may be addressed in the future. We compiled the errors into representative categories, which we define below and exemplify in Table I. In that table, we identify the system that predicted each example in parentheses, and indicate the specific component of the story in error in italics when applicable. We also discuss some general errors from papers about other visual storytelling approaches for which we were unable to obtain full working implementations.

4.5.1 Error Categories

We define our representative error categories as follows:

- **Grammatical Errors:** Incorrect use of verbs and tenses and/or subject-verb disagreements.
- **Contradictions:** Presence of inconsistent ideas within the same story (e.g., two sub-stories that are the opposite of each other).

- **Repetitions:** These errors were further subdivided into the following categories.
 - **Repetitions within Story:** Recurrence of the same sentence(s) within a story.
 - **Repetitions within Sentence:** Recurrence of the same phrase(s) within a sub-story.
 - **Repetitive Subject:** The sub-stories have the same subject and differ only in the adjective used to describe it.
 - **Repetitive Sentence Structure:** Most sentences start with “the [noun] was/were/is [adjective].” This leads to monotonous and unoriginal stories. We observed this error only in stories predicted by GLACNet.
- **Description in Isolation:** Most sub-stories start with “This is a picture of...” Sentences of this nature sound more like single image captions than contextual stories.
- **Singular/Plural Disagreement:** The same story has one sentence with a singular noun and another sentence with the same noun but in plural form.
- **Ghost Entities:** Some sub-stories make use of a pronoun that has no antecedent at all (e.g., referring to a new person who was not introduced formally in the preceding sub-stories). This leads to confusion.
- **Personification:** The attribution of human-like qualities to something non-human due to lack of common sense knowledge.
- **Absurdity:** Nonsensical stories or sub-stories.
- **Incomplete Stories:** Stories that have less than 5 sentences.

- **Point-of-View Inconsistency:** The narrative point of view randomly changes within the story (e.g., first person to second person), creating confusion.
- **Excessive Paraphrasing:** Presence of sub-stories that have similar meanings but are expressed using different words or phrases.

In addition to analyzing errors in stories with low predictions, we uncovered several anomalies in stories with high predictions. We provide examples of these in Table IV.

TABLE I. Example stories associated with each error category. This table is taken from Modi and Parde, 2019 [1].

Error Type	Example
Grammatical Errors	<i>there was a lot of people at the convention center</i> . we saw a lot of interesting signs . there were a lot of people there . there were a lot of people there . we had a great time at the bar . (AREL)
	the man was taking a walk on the sidewalk . he saw a lot of cool buildings . he saw a statue of a woman . <i>he was a big group of people</i> . he went to the museum . (GLACNet)
Contradictions	we went to the art gallery . we saw a lot of people there . <i>the streets were empty</i> . <i>the streets were full of people</i> . this is a picture of a woman . (AREL)
Repetitions within Story	<i>the bride and groom were very happy to be married</i> . the bride and groom were so happy to be married . <i>the bride and groom were so happy to be married</i> . we all had a great time at the reception . they danced the night away . (AREL)
	the family went to the zoo . they had a lot of fun . they were all very excited . <i>we had a great time</i> . <i>i had a great time</i> . (Contextualize, Show and Tell)
Repetitions within Sent.	it was a beautiful day for a trip to the beach . we took a trip to the beach . we went to the beach . the beach was beautiful . as <i>the sun went down</i> , <i>the sun went down</i> . (AREL)
Repetitive Subject	the water was calm and clear . the buildings were empty . the building was very tall . <i>the architecture was amazing</i> . <i>the architecture was breathtaking</i> . (GLACNet)
Repetitive Sentence Structure	the city is very beautiful . the bridge is amazing . the water is so nice . the ferris wheel is very good . the view is spectacular . (GLACNet)

TABLE II. Example stories associated with each error category. (TABLE I CONTINUED..)

Error Type	Example
Description in Isolation	<i>this is a picture of a street</i> . it was a long drive . there was a lot of damage to the side of the road . <i>this is a picture of a man</i> . after that we found a trail that was in the middle of the forest . (BLEU-RL)
	the flowers were very pretty the flowers were so beautiful . the flowers were beautiful . <i>this is a picture of a column</i> . it was a very nice place to be .(Contextualize, Show and Tell)
Singular/Plural Disagree- ment	the resort was beautiful . <i>the beach was nice</i> . <i>the beaches were amazing</i> . the water was so calm . the food was delicious . (GLACNet)
Ghost Enti- ties	the lady was smiling for the camera . she was excited to be there . she was having a good time . <i>she was so happy to see her</i> . she was looking at the car (GLACNet)
Personification	<i>the plane was very excited to be at the location</i> . the first stop was the train station . the guide was also impressed with the organization organization . the students were able to see the exhibits from the city . the entire group was so happy to be there . (GLACNet)
Absurdity	<i>the kitchen was a lot of work</i> . <i>here is a picture of a box</i> . <i>i had to take a picture of my work</i> . <i>we had to take a picture of the menu</i> . <i>i had a great time</i> . (AREL)
	<i>i bought a new car</i> . <i>this is a picture of a cat</i> . she was very excited . and i 'm so excited . this is my favorite gift . (GLACNet)

TABLE III. Example stories associated with each error category. (TABLE II CONTINUED)..

Error Type	Example
Incomplete Stories	i love to travel i had a great time . she is having a great time . we went to the city to see some of the people . i had a great time . (AREL)
Point-of-View Inconsistency	<i>i was so excited to be graduating today . he was very proud of his graduation . graduation day is always a success . he was very proud of his accomplishments . he was very proud of his accomplishments . (AREL)</i>
Excessive Paraphrasing	we went on a trip to location . there were a lot of interesting things to see . there <i>were many different kinds of</i> fruits and vegetables . <i>there was also a variety of</i> fruits and vegetables . i had a great time there . (AREL)
	we took the kids to the park . <i>we had a lot of fun . we had a great time .</i> the kids were having a great time . we had a great time . (Contextualize, Show and Tell)

TABLE IV

Example scoring anomalies, including the anomalous scores for each story.

Anom.	Example	Scores
Good Story, Low Score	we went to a halloween party . there were a lot of interesting things to see . we saw a lot of cool things . we saw a lot of old buildings . the christmas tree was the best part of the day . (AREL)	CIDEr: 4.27, BLEU-4: 0.00, BLEU-3: 15.79, BLEU-2: 29.76, BLEU-1: 50.95, ROUGE-L: 24.43, METEOR: 24.42
	the couple was excited to be on vacation . they were going to the mountains . they went down the road . they saw a beautiful church . they had a nice dinner . (GLACNet)	CIDEr: 0.62
Bad Story, High Score	the group of friends decided to go on a trip . they saw many interesting things . they stopped at a local restaurant . they had a great time . they ended up buying a new car . (GLACNet)	METEOR: 19.52, Bleu-4: 0.00, Bleu-3: 8.93, Bleu-2: 16.00, ROUGE-L: 22.55
	i went to a wedding last week . i had to take a picture of this beautiful flower . this is a picture of a woman . the flowers were so beautiful . the flowers were so beautiful . (AREL)	CIDEr: 20.90, Bleu-1: 71.79, Bleu-2: 43.47, METEOR: 33.98

TABLE V

Frequency (in terms of overall percentage) of the most common error types across all 1010 generated test stories by AREL and GLACNet and 1938 generated test stories by Contextualize, Show and Tell. This table is taken from Modi and Parde, 2019 [1].

Error Category	AREL-s-50	GLACNet	Contextualize, Show and Tell
Repetition of Sub-Stories	19.70%	2.08%	15.42%
Description in Isolation	29.01%	0%	15.79%

4.5.2 Discussion

The most common error types we observed were repetitions and descriptions in isolation; we present statistics indicating the frequencies of these errors for AREL, GLACNet, and Contextualize, Show and Tell in Table V (note that both occurred with the highest frequency in AREL). The rarest error category was that containing incomplete stories. This error appeared only in AREL stories, and only in three of the 1010 generated stories (0.003%).

The prevalence of repetitions in AREL is likely a side-effect of the model’s architecture—it generates the sub-stories for the whole album in parallel, rather than keeping track of what

was generated in the previous sub-story. We found that this structure also led to some stories having contradictory sentences. In contrast, GLACNet stories exhibited few repetitions because of the post-processing step employed after decoding. In this step, words for a sentence are sampled from a word probability distribution one hundred times and the most frequent word is selected. The words which occur in the generated sentences are also counted and the selection probabilities of words are decreased as their frequency increases.

It is somewhat surprising that the stories generated using Contextualize, Show and Tell also exhibited such a high frequency of repetitions, in spite of the fact that the model generated sub-stories sequentially. This demonstrates that some sort of feedback mechanism incorporating the model’s previously generated sub-stories is needed. The output of each of the five decoders in Contextualize, Show and Tell should be fed into the next decoder to keep track of previously generated sub-stories.

We observed that there were very few grammatical errors in the GLACNet stories, as the probabilities associated with function words (e.g., prepositions and pronouns) remained unchanged even if their rate of occurrence was high. In contrast, stories generated by AREL (which includes no such grammar-checking mechanism) included a considerable number of grammatical errors. GLACNet’s post-processing step still could be improved upon—we were somewhat surprised to find that some of its stories used both singular and plural forms of the same noun within a story. We assume the error occurred due to the fact that the model decreases the probability of frequently occurring words. Thus, if the singular noun occurred in the previous sub-story, its plural form gets included in the next sub-story.

The within-sentence repetitions may at least partially be a consequence of the presence of repetitions in some VIST training stories. In our analysis of the crowdsourced dataset we found that human typing/grammar errors were a relatively common occurrence, resulting in imperfect training data. Although the stories generated by GLACNet did not often exhibit repetitions due to the reasons mentioned in the paragraph above, there was a trade-off in terms of originality of the generated stories. We found that most were monotonous, using similar sentence structures for every story.

Descriptions in isolation, the single most prevalent error type we identified in AREL and Contextualize, Show and Tell stories, read more like image captions (describing the image’s contents) than components of a sequential story. We are perplexed as to why these errors were so common, since to the best of our understanding the models did not include any DII instances in their training sets. It may be the case that caption-like sub-stories are learned to be “safer” choices by these models, and thus generated more often than riskier contextual sub-stories.

Sentences that are lexically different but semantically similar cause redundancies in the story and are a common occurrence in both GLACNet and AREL. Since images in a sequential album are often visually similar to one another, it may be the case that both models predict that two (or more) images in a sequence refer to the same content. In attempting to vary the resulting sub-stories nonetheless, they succeed only at generating paraphrases of one another.

4.6 Recommendations

As evidenced by our error analysis, there is substantial scope for improvement in visual storytelling. Based on our observations, we make the following recommendations. First, **auto-**

matically preprocessing the DII and SIS training files remains an unexplored but **potentially highly useful preliminary step in the story generation process**. Doing so could aid future systems in avoiding grammatical mistakes, particularly if coupled with a post-processing mechanism similar to what is currently employed by GLACNet. Second, in terms of the post-processing mechanism itself, **incorporating temporal sequencing methods will yield more well-organized and coherent stories**. This could be done by sorting a (presumably jumbled) set of sub-stories after they have been generated, as was done by Agrawal et al. [14].

Third, it is common for current models to generate all sub-stories in parallel. This leads to repetitions and redundancies in the generated stories. **Modifying the architecture in such a way that the sub-stories are generated sequentially and the word tokens of the previously generated sub-stories are passed back to the model may lead to numerous benefits**. For instance, this feedback could be used to identify past sub-story topics, as well as to ensure that the singularity/plurality of subjects remains the same across the entire story. Incorporating a memory mechanism could also lessen the frequency of point-of-view inconsistencies, excessive paraphrasing, and contradictions. The architecture of the decoder used by Venugopalan et al. [29] can also be adopted for providing feedback at the word level along with the sub-story level feedback. This will help in keeping track of the previously generated words in the story and prevent in-sentence repetitions.

Fourth, **traditional image captions (DIIs) can be (carefully) leveraged to support the generation of high-quality stories**, for instance by facilitating named entity recognition

and thereby decreasing the frequency of ghost entities. Another way to avoid ghost entities is to (fifth) **incorporate a bottom-up and top-down visual attention mechanism**, such as that used in prior image captioning work [30], to learn image-specific features and facilitate visual grounding. Few-shot learning methods to jointly encode the images and text [31] could also be used in this regard. Matusov et al. [32] use a neural machine translation model which contains a visual encoder and a textual encoder, thus giving attention independently to both image features and source sentences. This technique is a more viable option. Finally, the anomalies we uncovered in our error analysis validate the position first put forward by Wang et al. [2], that automatic metrics leave much to be desired in terms of judging visual storytelling approaches. We recommend that a standardized human (or at least very humanlike) evaluation metric be included in the assessment of these approaches in the future.

CHAPTER 5

THE PROPOSED MODEL

5.1 Methods

Following our error analysis, we aimed to develop an improved model for visual storytelling. We select HCBNet [4] as the base model upon which we build our network, primarily due to its existing use of DII from the VIST dataset; as highlighted in our earlier analysis, doing so may offer particularly high utility in improving model quality. This hypothesis is supported by the performance results produced by HCBNet. An important addition in our proposed model is that we create a joint embedding of the image and text description by applying a dynamic coattention mechanism similar to that used by Xiong et al. [33] to focus on relevant parts of both. As human evaluation is not always possible to score the generated visual stories (particularly during model development, when collecting human scores would be impractical), here we experiment with BertScore [5] as an additional automated metric for the task. In the following sections we provide a detailed description of our network architecture.

5.2 Natural Language Generation

Natural Language Generation (NLG) is the process of automatically producing clear, sensible, and meaningful natural language phrases, sentences, and texts. Visual storytelling is at its core an NLG task that seeks to generate humanlike narratives for images. Other common NLG tasks include text summarization, image captioning, and text completion, among other

things. Although both rule-based and statistical methods have proven useful for NLG in the past, the dominant current direction makes use of artificial neural networks in various forms. As we make heavy use of artificial neural networks in our proposed model, we describe some key aspects of these models in the next few subsections.

5.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are fundamentally comprised of lots of tiny, interconnected processing units commonly referred to as “neurons.” These neurons are inspired by (although do not necessarily functionally mirror) the neurons of our brain. They can be used to solve a variety of complex, nonlinear problems. When many layers of such neurons are connected together, they form ANNs. ANNs then serve as universal functional approximators, in the sense that they can in theory learn to map any inputs to any outputs. They need to be trained thoroughly to do so for their respective tasks by tuning *hyperparameters*, or modifiable characteristics of their architecture.

Neural networks with many hidden layers and many neurons in each layer are called Deep Neural Networks (DNNs). DNNs are special because they can automatically generate useful features from unstructured data. In contrast, in other machine learning algorithms you specifically need to select the important features which will help in predicting and/or classifying the input. The downside of deep learning is that the networks require huge amounts of data to train properly, thus resulting in correspondingly huge processing times with many computations. DNNs also function as “black boxes”—data is fed in, and results are passed out, but it

is often unclear exactly why or how those results are obtained. Thus, another downside is that DNNs are less interpretable than other classification models.

The DNNs we will be using in this work are *Convolutional Neural Networks* and a type of *Recurrent Neural Networks* called *Long Short Term Memory Units* (LSTMs).

5.4 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) make use of sequential data. Consider a sequence of data X_0, X_1, \dots fed as input to an RNN. The RNN sees X_0 at time 0 and correspondingly performs its weight updates, similarly to a simple feedforward neural network. However, at timestep 1, it uses the hidden state of previous step (timestep 0) along with X_1 as input. The process continues for each timestep in the data sequence. In this way, the information (hidden representation) learned by the network is passed over time repeatedly.

The problem with RNNs is that they cannot remember information over long sequences, focusing more heavily on recent informations. They also suffer from the *vanishing gradient* problem: As the loss is back-propagated over several time steps, the gradient gets smaller and smaller until it has a negligible effect on weight updates.

5.5 Long Short Term Memory

In order to deal with the above issues, a variation of RNNs called Long Short Term Memory (LSTM) is used. They contain a memory cell C , whose gradient is exactly 1. This solves the vanishing gradient problem and also captures long distance dependencies.

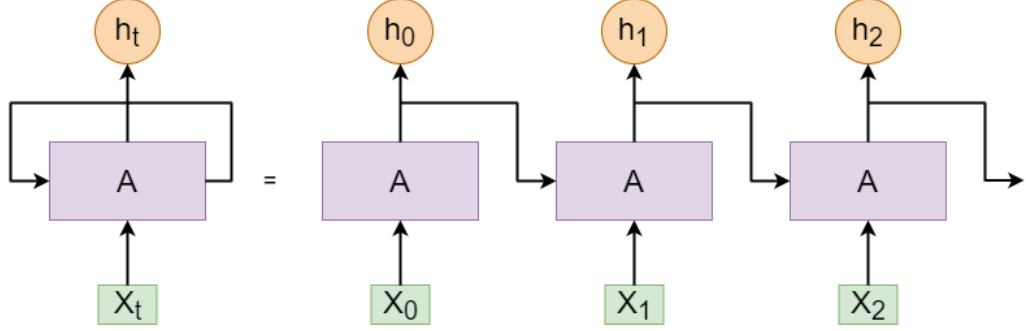


Figure 2. RNN with its unrolled version through time

LSTM uses a gating mechanism to do this. The detailed architecture is shown in Figure 3.

$$\mathbf{u}_{<t>} = \tanh(W_{<xu>}\mathbf{x}_{<t>} + W_{<hu>}h_{<t-1>} + \mathbf{b}_{<u>}) \quad (5.1)$$

$$\mathbf{i}_{<t>} = \sigma(W_{<xi>}\mathbf{x}_{<t>} + W_{<hi>}h_{<t-1>} + \mathbf{b}_{<i>}) \quad (5.2)$$

$$\mathbf{o}_{<t>} = \sigma(W_{<xo>}\mathbf{x}_{<t>} + W_{<ho>}h_{<t-1>} + \mathbf{b}_{<o>}) \quad (5.3)$$

$$\mathbf{C}_{<t>} = \mathbf{i}_{<t>} \odot \mathbf{u}_{<t>} + \mathbf{C}_{<t-1>} \quad (5.4)$$

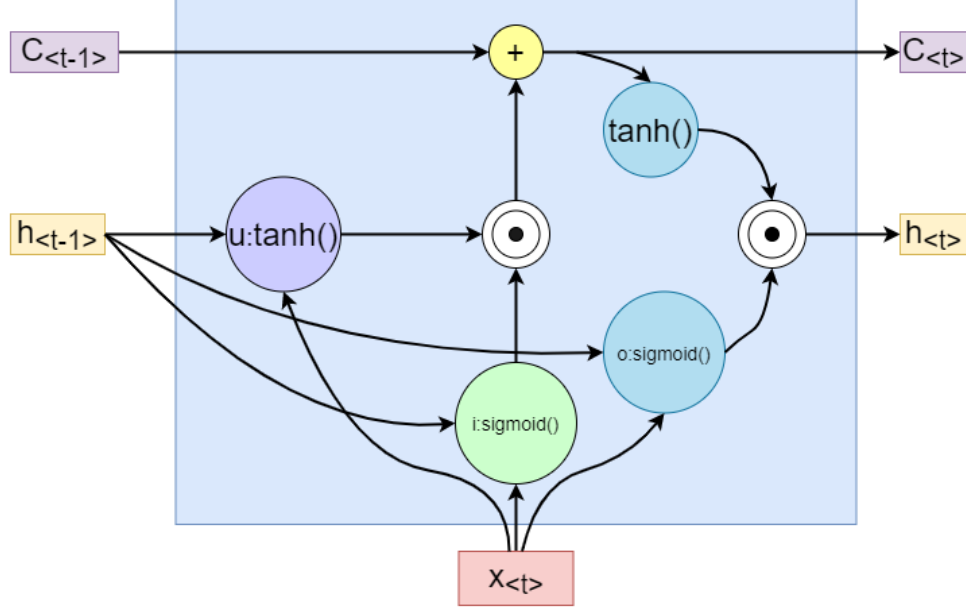


Figure 3. LSTM cell with update, input and output gates

$$\mathbf{h}_{<t>} = \mathbf{o}_{<t>} \odot \tanh(\mathbf{C}_{<t>}) \quad (5.5)$$

where, $\mathbf{x}_{<t-1>}$ and $\mathbf{x}_{<t>}$ are the inputs at the previous and current timesteps to the LSTM respectively, and $\mathbf{h}_{<t-1>}$ and $\mathbf{h}_{<t>}$ are the hidden states of the previous and current timesteps respectively.

Equation 6.1 [34] is the same as the normal update $\mathbf{u}_{<t>}$ of a simple RNN at a timestep t . The input gate $\mathbf{i}_{<t>}$ and output gate $\mathbf{o}_{<t>}$ are shown in Equations 6.2 and 6.3 respectively.

Both the gates specify subsets of information that is allowed to pass through them to the next stage. All the equations have been referred from Neubig, 2017 [34].

Although LSTMs already reduce some of the issues experienced with simple RNNs, they are able to capture only the previous context and not the future context. In order to train a neural language model that is good at predicting the next word, both future and previous contexts are needed. To address this need, Schuster and Paliwal [35] proposed bidirectional RNNs. Bidirectional RNNs combine two separate hidden RNN layers that read the same sequence in opposite directions to generate the output. In our model, we employ bidirectional LSTMs.

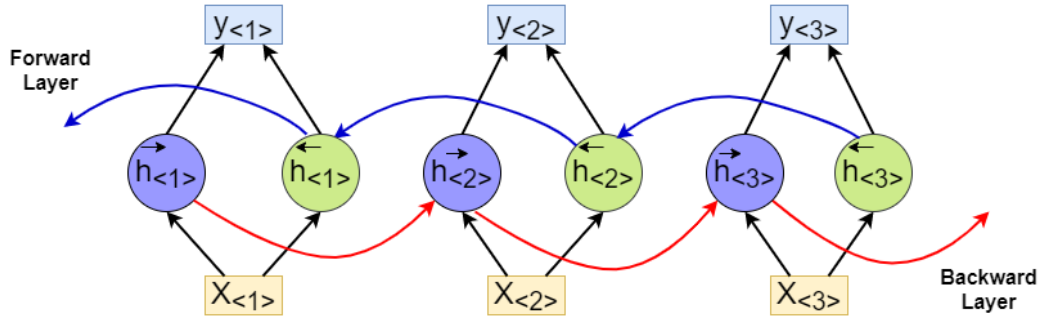


Figure 4. Sequence Processing in a Bidirectional RNN

$$\vec{h}_{<t>} = \sigma \left(W_{<\vec{h}x>} x_{<t>} + W_{<\vec{h}\vec{h}>} \vec{h}_{<t-1>} + b_{<\vec{h}>} \right) \quad (5.6)$$

$$\overleftarrow{h}_{<t>} = \sigma \left(W_{<\overleftarrow{h} x>} x_{<t>} + W_{<\overleftarrow{h} \overleftarrow{h}>} \overleftarrow{h}_{<t+1>} + b_{<\overleftarrow{h}>} \right) \quad (5.7)$$

$$y_{<t>} = W_{y\vec{h}} \vec{h}_{<t>} + W_{<y \overleftarrow{h}>} \overleftarrow{h}_{<t>} + b_y \quad (5.8)$$

More specifically, bidirectional LSTMs calculate the input in the forward direction to get the forward hidden sequence as shown in Equation 6.6. They also calculate the input in the backward direction to get the backward hidden sequence as shown in equation 6.7. $y_{<t>}$ is the encoded vector which is the concatenation of $\vec{h}_{<t>}$ and $\overleftarrow{h}_{<t>}$ expressed as $[\vec{h}_{<t>}; \overleftarrow{h}_{<t>}]$. The output sequence of the first hidden layer is shown in Equation 6.8. These equations have been taken from CS 224d lecture notes of Stanford University. This process is illustrated in Figure 4.

5.6 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are made up of *convolutional* layers, which are specialized hidden layers that compute functions over regions of input. They are mostly used for extracting high level features from images but can be used for text also. CNNs are powerful tools that are good at identifying local, fine grained patterns from large inputs, producing fixed-size output representations. Thus, they are very good at encoding images.

CNNs are composed of the following different types of layers:

- **Convolution:** This operation requires two types of signals. One is the input image, and the other is the filter matrix or kernel. Different sizes of filters can be applied on an

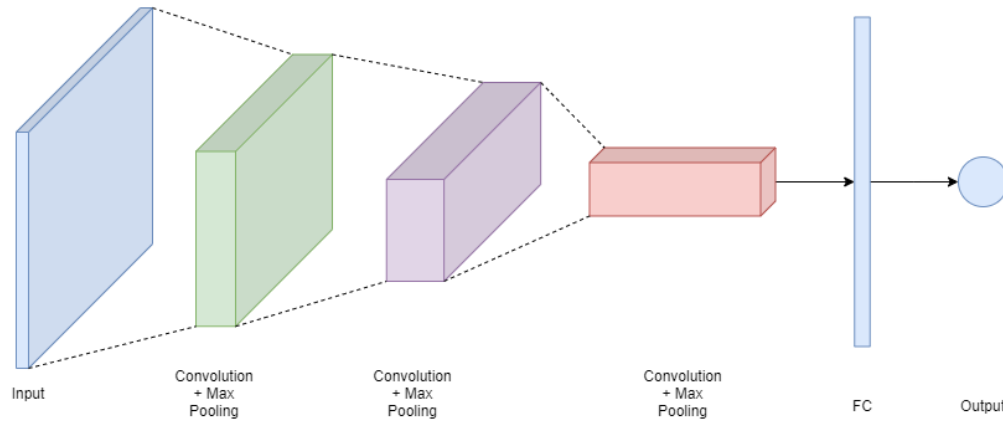


Figure 5. An example of a Convolutional Neural Network with convolution, pooling and fully connected (FC) layers.

input image to produce outputs of various sizes. The filter slides over the input image, and a simple dot product of the filter and the part of the input image it coincides with is taken. This is done repeatedly across the whole image to produce a final feature map. This process is known as a convolution. The weights for the filter have to be initialized randomly and are adjusted over time as the network learns. Including multiple types of convolutional layers in a model helps it to extract different types of features.

- **Pooling:** Pooling involves down sampling the feature maps generated from convolution, essentially reducing the dimensionality of the feature maps. Pooling does so by separating the feature map into bins and choosing only one value to represent the bin as a whole.

There are many types of pooling but the most common ones are min pooling and max pooling. As the name goes, max pooling chooses the maximum of all values in the bin to represent the bin and min pooling chooses the minimum one.

The output of a CNN is a long, dense representation in the form of a vector which encodes essential information about the input image or text.

5.7 Encoder-Decoder Networks

Encoder-decoder networks are commonly used in neural machine translation and sequence to sequence prediction. They take a sequence of input as the source, encode the source, and pass it to the decoder to generate the target data. Both the encoder and decoder are separate neural networks (typically recurrent neural networks). The encoder-decoder network encodes the information in the source into a vector representation (the final hidden state of the encoder) and passes this on to the decoder which decodes the encoded information into the target.

An advantage of such a network is that it can handle variable length input and outputs. It is also an end to end model which directly operates on the input source and target output. Thus, the loss function is optimized to learn the weights directly over input and output.

In the case of machine translation, we consider a text sequence in a source language as an input to the network. The network first generates an embedding of the input text. This embedding is then passed through the encoder (e.g., an LSTM). The final hidden state of the encoder is forwarded to the decoder (e.g., another LSTM) as input along with an embedding of the word generated in the previous iteration. In the initial case, a start of sentence <SOS> token is used to trigger the generation of words by the decoder. Using this information, the

decoder predicts the next word in the target language by calculating the probability by taking a softmax over its hidden state. It then continues to generate words until it encounters the end of sentence $\langle \text{EOS} \rangle$ token.

5.8 Hierarchical Context-Based Network

HCBNet is a hierarchical encoder-decoder network. The network is composed of two main parts: (1) A hierarchical encoder network and (2) a sentence decoder. We describe each of these components in more detail in the subsections below.

5.8.1 Hierarchical Encoder Network

5.8.1.1 Image Sequence Encoder

Within the hierarchical encoder network, the *Image Sequence Encoder* (ISE) component encodes the global context over all of the images it has seen at the current timestep. To do so, it first embeds each image from the input image sequence into a single vector representation using a CNN (specifically, a pretrained ResNet152 [36]). The extracted features are also passed through an LSTM network and the hidden state output of this LSTM is passed to the next LSTM to initialize its hidden state. This is repeated over the entire sequence and the final hidden state of the LSTM unit is passed through a fully connected layer to form the sequence embedding vector. This vector represents the global context of the sequence and is used to initialize the hidden state of the description encoder.

5.8.1.2 Image Description Encoder

The *Image Description Encoder* (IDE) contains two components: an *image encoder* and a *description encoder*.

Image Encoder (IE): The current image is passed through ResNet152 to extract features, which are then passed through an LSTM network. The hidden state of the LSTM at the current timestep is initialized by the hidden state of the LSTM from the previous timestep. The output of the LSTM at the current time step is passed through a fully-connected layer to generate the image embedding, which functions as the local context of the current image.

Description Encoder (DE): The theme type information incorporated in the image sequence embedding vector from the ISE is used as the initial hidden state of the first DE LSTM. The description is preprocessed into an embedding, which is then passed through an LSTM. The hidden state of the DE is passed through a fully-connected layer to generate the description embedding. This is then concatenated with the image embedding from above, and sent to the decoder. The final hidden state of the Sentence Decoder (SD) is concatenated with the DE’s hidden state. This concatenated vector is passed through a fully-connected layer to form the initial state of the SD. The current sentence theme is obtained from the hidden state of the DE and the previous hidden state of the DE tells the model what has been generated up until that point. Thus, the information extracted from the description is used to maintain temporal dependencies between the sentences.

5.8.2 Sentence Decoder

The Sentence decoder (SD) accepts as input the concatenation of the image embedding from the IE, the description embedding from the DE, and the previous word embedding. This is similar to a hard attention mechanism used to combine image and description contexts. The

hidden state of the SD of the current timestep is propagated to the next timestep. It uses the contexts from the IDE to then generate the visual narrative word by word.

5.9 Our Modification to HCBNet

The local features of the current image are learned from the image embedding vector, and the overall theme and local context are learned from the description embedding vector by the decoder. These two embeddings are thus learned from different vector spaces, and contain multimodal information. We hypothesized that adding an attention mechanism that combines the two modalities into a single space, or one which encodes the embedding of one modality into the space of another, would improve the HCBNet further. Our implementation and inclusion of this component forms the crux of our subsequent model experiments. Since the improved model will attend to both image and text features simultaneously, it will learn to focus on relevant parts from both.

Coattention was introduced by Lu et al. [37] as a mechanism for jointly reasoning about question and image attention while developing a visual question answering network. Visual question answering is another multimodal task which requires the model to form a correct understanding of both an image and a question, to correctly answer the question about an image. Lu et al. specifically employed a parallel co-attention and alternating co-attention between the question-image pairs. Their ablation studies show that the alternating coattention led to greater performance gains than the parallel one, and focuses on interpretable regions of images and questions while predicting the answer. They suggest that such a mechanism can be

adopted for other language and vision tasks as well; in our work, we test this conjecture. Our work is the first to employ a coattention mechanism for visual storytelling.

We adapt the coattention mechanism proposed by Xiong et al. [33], which originally sought to capture relations between questions and documents used in a text-based question answering system. We tweak the mechanism in such a way that it can instead be leveraged for image and description embeddings.

5.9.1 Coattention for Visual Storytelling

The image embedding obtained from the IE at the current timestep can be represented as $V \in R^{m \times l_1}$ where l_1 is the dimension of the image embedding vector. The description embedding from DE at the current timestep can be represented as $D \in R^{m \times l_2}$ where l_2 is the dimension of the word embedding. m is the maximum of all lengths of sub-stories in the full story (SIS).

Initially, an affinity matrix similar to Lu et al. [37] is calculated as follows:

$$L = \tanh(DW_bV^T) \quad (5.9)$$

where $W_b \in R^{l_2 \times l_1}$ contains the weights and is initialized randomly from a normal distribution. L is the affinity matrix which is the most important for calculating attention weights.

This matrix is normalized row wise to obtain attention weights $A^D \in R^{m \times m}$ across the image for each part of the description embedding, and column-wise to get attention weights $A^V \in R^{m \times m}$ across the description embedding for each part of the image.

$$A^D = \text{softmax}(L) \quad (5.10)$$

$$A^V = \text{softmax}(L^T) \quad (5.11)$$

$C_D \in R^{l_1 \times m}$ summarises or computes the attention contexts of the image in taking into account each part of the description embedding:

$$C_D = V A^D \quad (5.12)$$

The next two operations are performed in parallel. We also compute $D A^V$, the summaries of the description in light of each part of the image embedding. The summaries $C_D A^V$ compute the previous attention contexts in light of each part of the image embedding. $C_D A^V$ essentially represents the mapping of the description embedding space into the image embedding space.

$$C_V = [D; C_D] A^V \quad (5.13)$$

$C_V \in R^{(l_1 + l_2) \times m}$ is a co-dependent representation of the description and image, known as the coattention context. We use the notation $[x; y]$ for concatenating the vectors x and y horizontally.

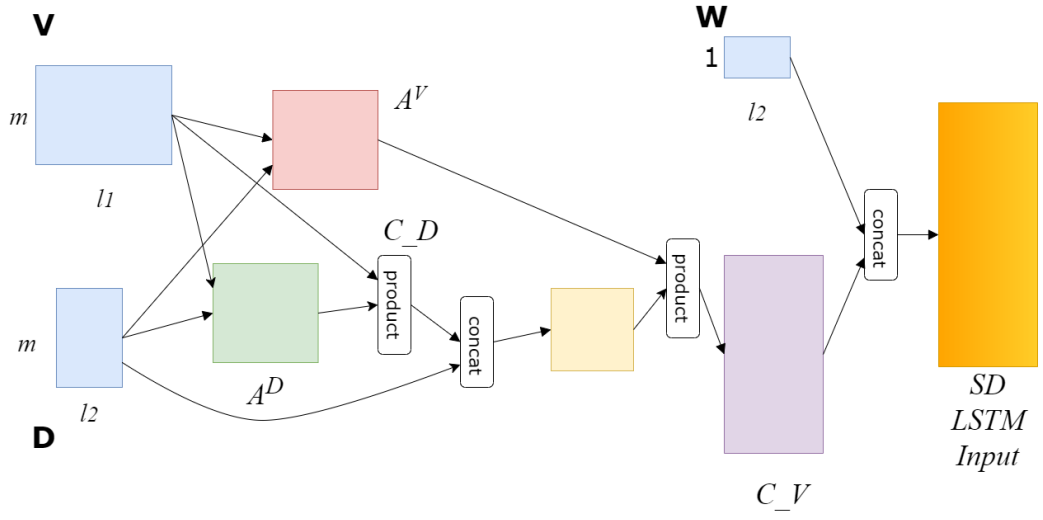


Figure 6. Coattention for Visual Storytelling. The affinity matrix L has been omitted but the attention weights A^D, A^V have been shown directly. W is the word embedding of the previously generated word by the SD.

The last step is to concatenate this coattention context with temporal information in the form of the word embedding of the previously generated word by the SD, so that they can be

fused together by passing them as input to the SD LSTM, which will then generate the next word.

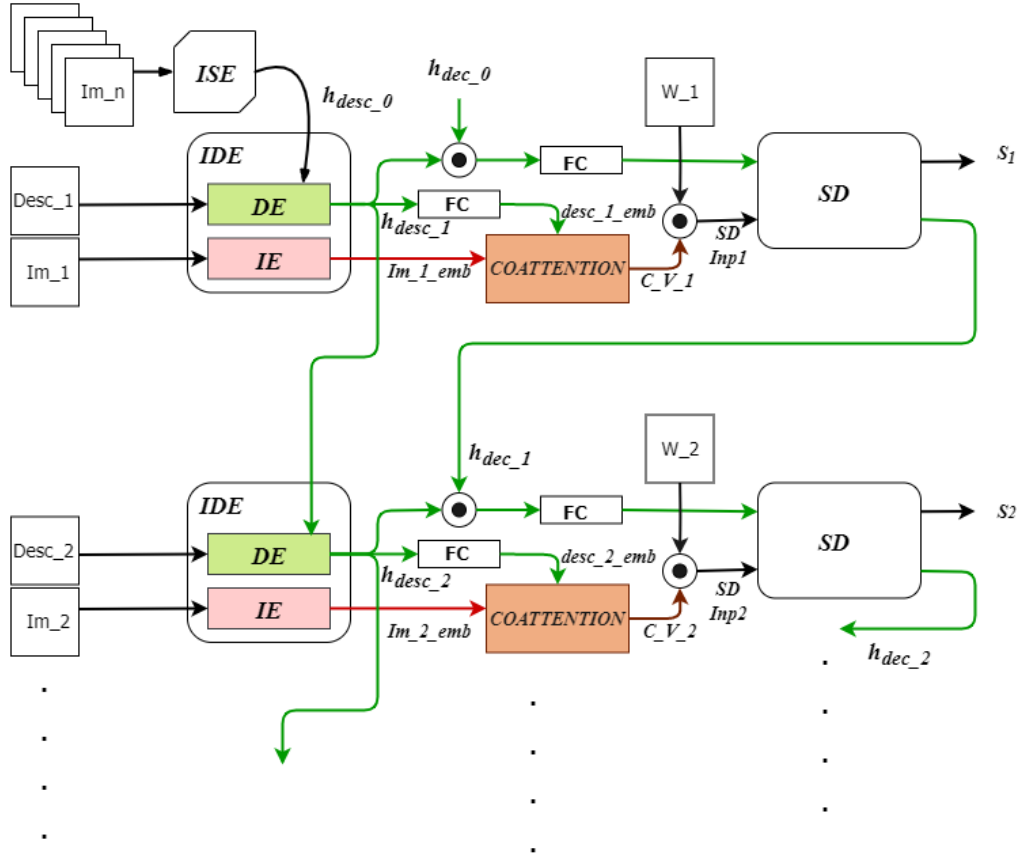


Figure 7. The Proposed Model with Coattention added to HCBNet (Nahian et al. 2019 [4]).

In Figure 7, Im_1, \dots, Im_n are the image inputs, $Desc_1, \dots, Desc_n$ are the corresponding descriptions for each image. h_{desc_0} is the image sequence embedding generated by the ISE. It represents the overall theme of the story and is used to initialize first hidden state (h_{desc_0}) of the DE. The output hidden state h_{desc_1} of the DE is used to initialize the DE of $Desc_2$ and so on. h_{dec_0} is the initial hidden state of the 1st SD. The output hidden state h_{desc_1} of this SD is concatenated with h_{dec_2} and passed through a fully connected (FC) layer and used to initialize the hidden state of SD. This is repeated for all images. A coattention module as described in Figure 6 which takes as input the current image embedding Im_i_emb and the current description embedding $desc_i_emb$ to produce CV_i . This attended vector is concatenated with the embeddings of previously generated word at each timestep and thus W_i represents those embeddings. This is used as input to the SD (SD Inpi) which generates a sentence Si , word by word.

CHAPTER 6

EXPERIMENTAL SETUP

To perform our experiments, we resize images in the VIST dataset to 256×256 pixel dimensions, and then transform them into 224×224 (the dimension needed for use as input to ResNet152) images via random cropping along with a horizontal flip while training. Each word is embedded into a 256-dimension vector. We remove stop words from the descriptions, but not from the stories.

We train the decoder LSTM using teacher forcing (the ground truth/ target is fed as input to the decoder LSTM while predicting the next word). The hidden layer size of both LSTMs is set to 1024. Two layers of bidirectional LSTMs are used, resulting in a output hidden state of size 2048. This output is then resized using a fully-connected layer. Batch normalization and dropout have been applied following this to prevent overfitting the network. Each fully-connected layer that follows an LSTM layer has batch normalization applied to it. A dropout of 0.5 is applied on each LSTM cell, and to the output vector of the sentence decoder.

The learning rate is set to 0.0001, while the weight decay is set to $1e-5$ for the Adam optimizer. The batch size is set to 64, and the model is trained for 40 epochs.

CHAPTER 7

EVALUATION AND RESULTS

7.1 Quantitative Evaluation

In order to quantitatively evaluate our model, we make use of a suite of automatic metrics, described in detail earlier in Section 4.4. We also experiment with another automated metric, BERTScore [5]. BERTScore was recently introduced as an improved mechanism for automatically evaluating generated text.¹

7.1.1 BERTScore

BERTScore ([5]) is an automatic evaluation metric for text generation tasks such as machine translation, image captioning, and text summarization. It computes a sum of the cosine similarity between the contextual BERT [38] embeddings of the tokens of the candidate and reference sentences. It is different from the other commonly used evaluation metrics because it doesn't perform hard string matching or text alignment checks. It is also better able to manage long distance dependencies.

BERTScore captures paraphrase matches well, because BERT encodes semantic meanings of words. It maximizes the matching similarity score by using a greedy strategy. Essentially, it tries to match each token from one sentence to the most similar token in the other sentence.

¹Although we originally spent some time trying to develop our own metric along similar lines, BERTScore was released first; thus, we decided to make use of it rather than investing additional time on our metric for the present.

BERTScore also makes use of importance weighing, which is based on the observation that rare words indicate more sentence similarity than common words. To do so, it makes use of inverse document frequency scores.

BERT-Score is fast, robust, easy to use and task agnostic. Unlike BLEU, it is not bound by n-gram matching. It can also capture dependencies over long ranges. These traits make it particularly suitable for our task; thus we use it here to evaluate our generated visual stories.

7.2 Results and Discussion

As can be seen from Table VI, AREL and CIDEr-RL have higher scores compared to our model for all standard metrics (METEOR, CIDEr, ROUGE-L, BLEU-1, BLEU-2, BLEU-3, and BLEU-4). As mentioned previously, such automatic metrics have been shown to exhibit low correlation with human evaluation scores for visual stories, in some cases even exhibiting a negative correlation [39]. Thus, we mainly provide these metrics here to facilitate comparison with prior work for which no other metrics were computed.

We note that we did observe a noticeable increase in the precision-based metrics scores (e.g., BLEU and CIDEr) for our model when compared to the original HCBNet without coattention. ROUGE-L also increases. AREL scores highest overall using these standard metrics, but as mentioned in our earlier analysis, we observed that its stories tend to contain many repetitions and produce more literal descriptions than actual human-like stories.

As a consequence of all the problems faced when evaluating stories with the classical automatic metrics above, we turn our focus instead to the obtained BERTScores for the remainder

TABLE VI. Performance as reported in the source papers [2, 3]. BLEU-RL, METEOR-RL, and CIDEr-RL were baseline reinforcement learning approaches using BLEU, METEOR, and CIDEr scores as their reward functions, respectively [2].

Model	METEOR	CIDEr	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
<i>AREL-s-50</i>	34.9	9.1	29.4	62.9	38.4	22.7	14.0
<i>BLEU-RL</i>	34.6	8.9	29.0	62.1	38.0	22.6	13.9
<i>CIDEr-RL</i>	34.9	8.1	29.7	61.9	37.8	22.5	13.8
<i>GLACNet</i>	30.14	3.7	28.2	53.4	29.4	15.6	8.6
<i>Contextualize, Show and Tell</i>	34.4	5.1	29.2	60.1	36.5	21.1	12.7
<i>HCBNet</i>	34.0	5.1	27.4	59.3	34.8	19.1	10.5
<i>Our Model</i>	31.1	5.5	27.9	60.0	35.6	19.9	11.1

of our quantitative evaluation. BERTScore has been shown to exhibit high correlation with human evaluation results for various text generation tasks.

Table VII shows the BERTScores for all core visual storytelling models considered. Since scores computed by BERTScore fall into a very small range when using a RoBERTa base [40], we report rescaled BERTScores to enhance their readability as directed by Zhang et al. [5].

TABLE VII. BERTScore Precision (P_{BERT}), Recall(R_{BERT}) and F1 Score $F1_{BERT}$. The maximum values are made bold.

Model	P_{BERT}	R_{BERT}	F_{BERT}
<i>AREL</i>	26.88	13.58	20.24
<i>GLACNet</i>	27.05	11.43	19.22
<i>Contextualize, Show and Tell</i>	23.20	9.6	16.41
<i>Our Model</i>	29.68	14.19	21.91

Our model achieves the highest BERTScore precision, recall, and F1 score compared to all other models. We note that for image captioning tasks, Zhang et al. [5] find P_{BERT} to be least useful and R_{BERT} to be most useful. As visual storytelling is somewhat similar to image captioning, we focus primarily on that outcome. Our model’s R_{BERT} score outperforms that of AREL, GLACNet, and Contextualize, Show and Tell by a percent increase of 4.5%, 24.1%, and 47.8%, respectively.

7.3 Qualitative Evaluation

Following our quantitative evaluation, we also qualitatively evaluate our generated stories by computing n-gram counts, average sentence length, and average story length, across various visual storytelling models. From Figure 8, we can see that stories predicted by our model contain more unique n-grams than observed in other models, providing evidence that our model generates more lexically diverse stories. This may be jointly attributable to our inclusion of

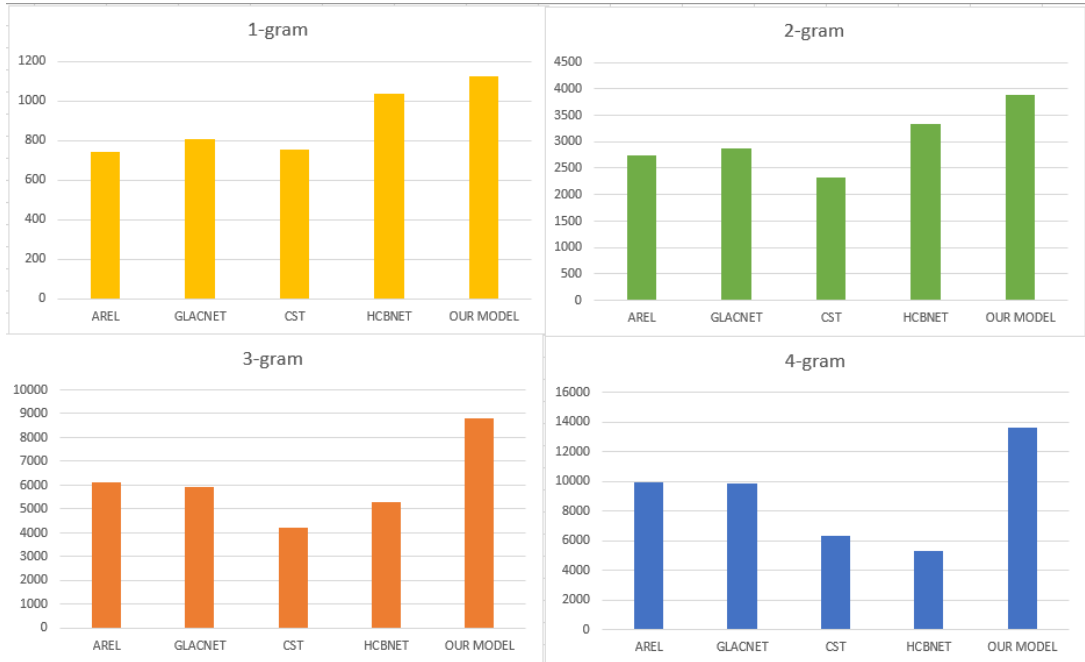


Figure 8. Histograms showing the count of unique 1-grams, 2-rams, 3-grams and 4-grams in the various models.

DIIs and our novel coattention mechanism when training our model. Overall, it is clear that the DIIs help in generating much higher quality stories than is observed in baselines that do not use them. The stark difference in n -gram counts of HCBNet and our model (particularly at higher values of n) also demonstrates that the attention mechanism effectively enabled the model to map from the image embedding space and the description embedding space, and thus highlight more relevant (and therefore often more unique) words from the descriptions used to generate these visual stories. We hypothesize that the the SD LSTM also played a role in this

increased story customization, by attempting to learn relations between the coattention vector C_V and the previously generated word embedding.

AREL produced very long visual stories, as we can see from Table VIII; however, its unique unigram count is much lower than all other models. This lends more concrete support to our earlier finding that the long AREL stories are riddled with repetitions and are in general of relatively low quality, despite achieving high scores from standard metrics like BLEU. We also note that GLACNet seems to be producing shorter stories. An earlier investigation by Hsu et al. [39] found that when humans were asked to manually edit GLACNet stories, their length generally increased to an average length of 36.7 words. Thus, (a) the increase in the size of stories generated by our model may be attributable to the fact that our stories are closer to “edited” stories,¹ and (b) out of all models, our average story length is closest to the average length of human-edited stories, lending an additional small but interesting similarity between stories generated by our model and human stories, relative to those generated by other models.

Finally, we counted the number of stories generated by our model which have repetitive sub-stories in them. Out of the 2019 stories generated, only 1 story has two repetitive sentences. This is a massive improvement from the other baselines, which have had a considerable percent of repetitive sub-stories as seen in Table V. This once again proves the efficacy of our model. Further, only 1.08% of the stories generated sub-stories resembling DII (e.g., “this is a picture of...”).

¹We remind readers that our model is built upon a variation of GLACNet.

TABLE VIII. Experimental results showing some word properties

Model	Average No. of words per sentence	Average No. of words per story
<i>AREL</i>	8.52	42.67
<i>GLACNet</i>	6.86	34.36
<i>Contextualize,</i> <i>Show and</i> <i>Tell</i>	8.75	33.86
<i>HCBNet</i>	6.141	30.7
<i>Our Model</i>	7.09	35.47



<p>Description : <i>a lot of bread and chips gathered in the grass . several ears of corn lie on two shopping bags on a grassy ground . a large white and clear plastic package with many hotdogs sitting in the grass . grill top with wrapped corn and cucumbers sliced up . a jar of red pepper flakes are labeled as being hot stuff .</i></p>
<p>AREL : <i>i bought a lot of ingredients for the barbecue . there were a lot of vegetables . this is the best part of the day . i made a lot of food . we had a great time .</i></p>
<p>GLACNet : <i>the market was very busy . there were many different kinds of foods . some of the food was delicious . the meat was also good . the menu was very tasty .</i></p>
<p>CST : <i>i went to the market yesterday . i made a lot of fruits . i also bought some snacks while i was there . the food was amazing the <UNK> was very nice .</i></p>
<p>OUR MODEL : <i>i went to the grocery store . there were many different kinds of food . some of them were very tasty . they were all very good . afterward i got a lot of stuff .</i></p>

Figure 9. The different visual stories generated by the baselines and our model.



Our Model : *the bride and groom were ready to start their wedding . they took a picture of the bridesmaids . then , they danced with a couple of friends . after that they had a few drinks . everyone was happy to be there .*

Figure 10. A very imaginative and good quality visual story generated by our model.

CHAPTER 8

CONCLUSION AND FUTURE WORK

In this work, we conducted a comprehensive error analysis of recent visual storytelling approaches. We note current shortcomings in this area, and make recommendations for addressing these limitations in future work. We find that the most common errors are repetitions, the presence of traditional image descriptions, and a lack of creativity in the machine-generated stories. Preprocessing the training text, developing a combined visual and text co-attention mechanism, and sequentially generating sub-stories and providing them as feedback to the model can all help to ameliorate these issues. Specifically, including these elements could help in the generation of more context-aware sequential sub-stories, and temporally sequencing the sub-stories will produce more creative, coherent, relevant, and most importantly, humanlike stories.

We made use of some of the aforementioned recommendations to develop our own improved model for visual storytelling. Specifically, we introduced a novel coattention mechanism between image embeddings and description embeddings for visual storytelling. This leads to the generation of a joint representation of the two modalities, a co-dependent vector which leads to the generation of more coherent, visually grounded, cohesive and much higher quality visual stories. We also experimented with a recently introduced metric called BERTScore to evaluate our stories. We achieve state of the art BERTScores for stories generated by our model. In the future, we plan to experiment with knowledge graphs and residual connections in the network in order to further enhance our model performance.

APPENDIX

COPYRIGHT PERMISSION

This study contains work from Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. [1]. I am the first author and my advisor Prof. Natalie Parde is the second author. The license for the paper is attached on page 59.

The chapter wise breakdown containing parts from previously published work is as follows:

CHAPTER 1: INTRODUCTION (Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.)


CHAPTER 2: RELATED WORK (Part of it is previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.)

CHAPTER 3: DATA (Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.)

APPENDIX (Continued)

CHAPTER 4: AN ANALYSIS OF ERRORS IN EXISTING VISUAL STORY-TELLING MODELS (Previously published as Modi, Y. and Parde, N.: The steep road to happily ever after: An Analysis of Current Visual Storytelling Models. In Proceedings of the Second Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.)

APPENDIX (Continued)



Attribution 4.0 International (CC BY 4.0)


This is a human-readable summary of (and not a substitute for) the [license](#). [Disclaimer](#).

You are free to:


Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.



Under the following terms:



Attribution — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable [exception or limitation](#).

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as [publicity, privacy, or moral rights](#) may limit how you use the material.

CITED LITERATURE

1. Modi, Y. and Parde, N.: The steep road to happily ever after: an analysis of current visual storytelling models. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language* , pages 47–57, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
2. Wang, X., Chen, W., Wang, Y.-F., and Wang, W. Y.: No metrics are perfect: Adversarial reward learning for visual storytelling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* , pages 899–909. Association for Computational Linguistics, 2018.
3. 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.
4. Nahian, M. S. A., Tasrin, T., Gandhi, S., Gaines, R., and Harrison, B.: A hierarchical approach for visual storytelling using image description. In *Interactive Storytelling* , eds. R. E. Cardona-Rivera, A. Sullivan, and R. M. Young, pages 304–317, Cham, 2019. Springer International Publishing.
5. Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y.: Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations* , 2020.
6. Huang, T.-H. K., Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R., He, X., Kohli, P., Batra, D., Zitnick, C. L., Parikh, D., Vanderwende, L., Galley, M., and Mitchell, M.: Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* , pages 1233–1239. Association for Computational Linguistics, 2016.
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* , 115(3):211–252, 2015.
8. Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A.: Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* , 19(2), 1993.

9. Banerjee, S. and Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* , pages 65–72, Ann Arbor, June 2005. Association for Computational Linguistics.
10. Gonzalez-Rico, D. and Pineda, G. F.: Contextualize, show and tell: A neural visual storyteller. *CoRR* , abs/1806.00738, 2018.
11. Hsu, C., Chen, S., Hsieh, M., and Ku, L.: Using inter-sentence diverse beam search to reduce redundancy in visual storytelling. *CoRR* , abs/1805.11867, 2018.
12. Lukin, S., Hobbs, R., and Voss, C.: A pipeline for creative visual storytelling. In *Proceedings of the First Workshop on Storytelling* , pages 20–32. Association for Computational Linguistics, 2018.
13. Yu, L., Bansal, M., and Berg, T.: Hierarchically-attentive rnn for album summarization and storytelling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* , pages 966–971, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
14. Agrawal, H., Chandrasekaran, A., Batra, D., Parikh, D., and Bansal, M.: Sort story: Sorting jumbled images and captions into stories. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* , pages 925–931, Austin, Texas, November 2016. Association for Computational Linguistics.
15. Jain, P., Agrawal, P., Mishra, A., Sukhwani, M., Laha, A., and Sankaranarayanan, K.: Story generation from sequence of independent short descriptions. In *Proceedings of the KDD Workshop on Machine Learning for Creativity* , 2017.
16. Yang, P., Luo, F., Chen, P., Li, L., Yin, Z., He, X., and Sun, X.: Knowledgeable storyteller: A commonsense-driven generative model for visual storytelling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* , pages 5356–5362. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
17. Wang, R., Wei, Z., Li, P., Zhang, Q., and Huang, X.: Storytelling from an image stream using scene graphs. 2019.

18. Hsu, C.-C., Chen, Z.-Y., Hsu, C.-Y., Li, C.-C., Lin, T.-Y., Huang, T.-H., and Ku, L.-W.: Knowledge-enriched visual storytelling. *ArXiv* , abs/1912.01496, 2019.
19. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* , 123(1):32–73, 2017.
20. Vedantam, R., Zitnick, C. L., and Parikh, D.: Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , pages 4566–4575, June 2015.
21. Jung, Y., Kim, D., Woo, S., Kim, K., Kim, S., and Kweon, I. S.: Hide-and-tell: Learning to bridge photo streams for visual storytelling. *CoRR* , abs/2002.00774, 2020.
22. Hu, J., Cheng, Y., Gan, Z., Liu, J., Gao, J., and Neubig, G.: What makes a good story? designing composite rewards for visual storytelling. *ArXiv* , abs/1909.05316, 2019.
23. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L.: Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014* , eds. D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, pages 740–755, Cham, 2014. Springer International Publishing.
24. Kullback, S. and Leibler, R. A.: On information and sufficiency. *The Annals of Mathematical Statistics* , 22(1):79–86, 1951.
25. He, K., Zhang, X., Ren, S., and Sun, J.: Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* , pages 770–778, 2016.
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z.: Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , pages 2818–2826, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.
27. Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics* , pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

28. Lin, C.-Y. and Och, F. J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 605–612, Barcelona, Spain, July 2004.
29. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R. J., Darrell, T., and Saenko, K.: Sequence to sequence - video to text. *CoRR*, abs/1505.00487, 2015.
30. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.
31. Dong, X., Zhu, L., Zhang, D., Yang, Y., and Wu, F.: Fast parameter adaptation for few-shot image captioning and visual question answering. In *Proceedings of the 2018 ACM on Multimedia Conference (ACM MM)*, 2018.
32. Matusov, E., Way, A., Calixto, I., Stein, D., Lohar, P., and Castilho, S.: Using images to improve machine-translating e-commerce product listings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 637–643, 2017.
33. Xiong, C., Zhong, V., and Socher, R.: Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016.
34. Neubig, G.: Neural machine translation and sequence-to-sequence models: A tutorial. *CoRR*, abs/1703.01619, 2017.
35. Schuster, M. and Paliwal, K.: Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November 1997.
36. He, K., Zhang, X., Ren, S., and Sun, J.: Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
37. Lu, J., Yang, J., Batra, D., and Parikh, D.: Hierarchical question-image co-attention for visual question answering. *CoRR*, abs/1606.00061, 2016.
38. Devlin, J., Chang, M., Lee, K., and Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

39. Hsu, T., Huang, C., Hsu, Y., and Huang, T. K.: Visual story post-editing. *CoRR* , abs/1906.01764, 2019.
40. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* , 2019.

VITA

NAME: YATRI MODI

EDUCATION: M.S., Computer Science, University of Illinois at Chicago,
Chicago, Illinois, Fall 2018 - ON.

B.Eng., Computer Engineering, University of Mumbai, Mumbai, Maharashtra, India, Fall 2014 - Spring 2018.

EXPERIENCE: Graduate Assistant, HRIS, University of Illinois at Chicago,
Spring 2019 - Spring 2020.

PUBLICATIONS: Modi, Y., and Parde, N. 2019. The steep road to happily ever after: An analysis of current visual storytelling models. In *NAACL Workshop on SiVL*, 47–57.