FORCED ATTENTION FOR IMAGE CAPTIONING

by

Hemanth Devarapalli

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

December 2018

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Julia Rayz, Chair

Department of Computer and Information Technology

Dr. John Springer

Department of Computer and Information Technology

Dr. Baijian Yang

Department of Computer and Information Technology

Approved by:

Dr. Eric Matson

Head of the Graduate Program

I dedicate my thesis to my family and friends. Special thanks go to my parents Katyayani and Bapineedu Chowdary, who have always been supportive and a source of strength during tough times.

ACKNOWLEDGMENTS

I would first like to thank my advisor Dr. Julia Rayz for her help and guidance through the past two and a half years. The weekly one on one meetings and bi-weekly lab gatherings have given me invaluable feedback and suggestions to work with. Furthermore, I am grateful to Dr. Yang and Dr. Springer for agreeing to be on my committee and giving me feedback.

I would also like to thank the statistical consulting service provided by the Purdue Stats department. Specifically, Dr. Bruce Craig, and PhD candidate Fan Wu for helping me test for statistical significance of my results.

I'm thankful to Mr. Kelton Render and Ms. Kristine Hoggatt, for the graduate assistantship I held during the entirety of my time at Purdue. Special thanks to Ms. Kristine Hoggatt for letting me have flexible hours at the assistantship.

I've been lucky to have had help and support from my uncle and aunt, Sai and Tanuja Devarapalli. Moving to a new country couldn't have been any easier. I'll always be indebted to them.

I wish to thank Manideep Pabba, one of the best people I've had the pleasure of being friends with. Thank you for all the help throughout the past few years.

I will always be grateful for the Department of Computer and Information Technology for admitting me to their masters program and giving me this opportunity to be involved with academic research.

Finally, I would like to thank my friends and family, for their continued support and confidence in me.

TABLE OF CONTENTS

LIST OF TABLES 8
LIST OF FIGURES 10
LIST OF ABBREVIATIONS 11
GLOSSARY 12
ABSTRACT 13
CHAPTER 1. INTRODUCTION 14
1.1 Statement of the Problem 15
1.2 Research Question 15
1.3 Scope 15
1.4 Significance 16
1.5 Assumptions 17
1.6 Limitations 17
1.7 Delimitations 17
CHAPTER 2. LITERATURE REVIEW 19
2.1 Introduction 19
2.2 Early Attempts 19
2.3 Advent of new Datasets 20
2.4 Deep Learning Techniques 22
2.4.1 Usage of Deep Learning Techniques for Text Tasks 22
2.4.2 Deep Learning Techniques for Image analysis and Classification 24
2.4.3 Image Captioning with Deep Learning 26
2.5 Further Innovations in Image Captioning 28
2.6 Advent of Attention Models 31
2.7 Molding the Attention Models 34
2.8 Evaluation Metrics 37
2.9 Summary 39
CHAPTER 3. FRAMEWORK AND METHODOLOGY 39
3.1 Research framework 39
3.1.1 Network architecture 39

3.1.2 Modification of Attention 44
3.2 Dataset 45
3.3 Evaluation 46
3.4 Testing Methodology 46
3.4.1 Pruning the Dataset 46
3.4.2 Training the neural network 46
3.4.3 Generation of the caption 47
3.4.4 Selecting the Focus Objects 47
3.4.5 Establishing the baselines 48
3.4.6 Extracting attention map for the Focus Object 48
3.4.7 Feeding the attention map to the Attention mechanism 48
3.4.8 Evaluating the forced attention neural network 49
3.5 Summary 49
CHAPTER 4. RESULTS AND DISCUSSIONS 50
4.1 Focus Object: Dog 50
4.1.1 Results for Focus Object: Dog with Static Forced Attention 50
4.1.2 Results for Focus Object: Dog with Gradual Forced Attention 51
4.2 Focus Object: Pizza 53
4.2.1 Results for Focus Object: Pizza with Static Forced Attention 53
4.2.2 Results for Focus Object: Pizza with Gradual Forced Attention 54
4.3 Focus Object: Frisbee 56
4.3.1 Results for Focus Object: Frisbee with Static Forced Attention 57
4.3.2 Results for Focus Object: Frisbee with Gradual Forced Attention 57
4.4 Focus Object: Clock 60
4.4.1 Results for Focus Object: Clock with Static Forced Attention 60
4.4.2 Results for Focus Object: Clock with Gradual Forced Attention 60
4.4.3 Discussion for Focus Object: Clock 63
4.5 Focus Object: Train 66
4.5.1 Results for Focus Object: Train with Static Forced Attention 66
4.5.2 Results for Focus Object: Train with Gradual Forced Attention 66
4.5.3 Discussion for Focus Object: Train 69

4.6 Focus Object: Toilet 71
4.6.1 Results for Focus Object: Toilet with Static Forced Attention 71
4.6.2 Results for Focus Object: Toilet with Gradual Forced Attention 71
4.6.3 Discussion for Focus Object: Toilet 74
4.7 Significance of the improvements 77
CHAPTER 5. CONCLUSION AND FUTURE WORK 78
5.1 Future Work 78
5.2 Final Words 79
REFERENCES 80

LIST OF TABLES

Table 1: Metrics for focus object - dog with static forced attention 50
Table 2: Metrics for focus object – dog with gradual forced attention 51
Table 3: BLEU - 1 metric for baseline and gradual forced attention arch.....51
Table 4: BLEU - 2 metric for baseline and gradual forced attention arch.....52
Table 5: BLEU - 3 metric for baseline and gradual forced attention arch.....52
Table 6: BLEU - 4 metric for baseline and gradual forced attention arch.....52
Table 7: METEOR metric for baseline and gradual forced attention arch. 52
Table 8: ROGUE L metric for baseline and gradual forced attention arch. 53
Table 9: CIDer metric for baseline and gradual forced attention arch. 53
Table 10: Metrics for focus object – pizza with static forced attention 54
Table 11: Metrics for focus object – pizza with gradual forced attention 54
Table 12: BLEU - 1 metric for baseline and gradual forced attention arch.....55
Table 13: BLEU - 2 metric for baseline and gradual forced attention arch.....55
Table 14: BLEU - 3 metric for baseline and gradual forced attention arch.....55
Table 15: BLEU - 4 metric for baseline and gradual forced attention arch.....55
Table 16: METEOR metric for baseline and gradual forced attention arch. 56
Table 17: ROGUE L metric for baseline and gradual forced attention arch. 56
Table 18: CIDer metric for baseline and gradual forced attention arch. 56
Table 19: Metrics for focus object – frisbee with static forced attention 57
Table 20: Metrics for focus object – frisbee with gradual forced attention.....57
Table 21: BLEU - 1 metric for baseline and gradual forced attention arch.....58
Table 22: BLEU - 2 metric for baseline and gradual forced attention arch.....58
Table 23: BLEU - 3 metric for baseline and gradual forced attention arch.....58
Table 24: BLEU - 4 metric for baseline and gradual forced attention arch.....59
Table 25: METEOR metric for baseline and gradual forced attention arch. 59
Table 26: ROGUE L metric for baseline and gradual forced attention arch. 59
Table 27: CIDer metric for baseline and gradual forced attention arch. 59
Table 28: Metrics for focus object – clock with static forced attention 60
Table 29: Metrics for focus object – clock with gradual forced attention 61

Table 30: BLEU - 1 metric for baseline and gradual forced attention arch.....61
Table 31: BLEU - 2 metric for baseline and gradual forced attention arch.....61
Table 32: BLEU - 3 metric for baseline and gradual forced attention arch.....62
Table 33: BLEU - 4 metric for baseline and gradual forced attention arch.....62
Table 34: METEOR metric for baseline and gradual forced attention arch. 62
Table 35: ROGUE L metric for baseline and gradual forced attention arch. 62
Table 36: CIDer metric for baseline and gradual forced attention arch. 63
Table 37: Metrics for focus object – train with Static Forced Attention 66
Table 38: Metrics for focus object – train with gradual forced attention 67
Table 39: BLEU - 1 metric for baseline and gradual forced attention arch.....67
Table 40: BLEU - 2 metric for baseline and gradual forced attention arch.....67
Table 41: BLEU - 3 metric for baseline and gradual forced attention arch.....68
Table 42: BLEU - 4 metric for baseline and gradual forced attention arch.....68
Table 43: METEOR metric for baseline and gradual forced attention arch. 68
Table 44: ROGUE L metric for baseline and gradual forced attention arch. 68
Table 45: CIDer metric for baseline and gradual forced attention arch. 69
Table 46: Metrics for focus object – toilet with static forced attention.....71
Table 47: Metrics for focus object – toilet with gradual forced attention 72
Table 48: BLEU - 1 metric for baseline and gradual forced attention arch.....72
Table 49: BLEU - 2 metric for baseline and gradual forced attention arch.....72
Table 50: BLEU - 3 metric for baseline and gradual forced attention arch.....73
Table 51: BLEU - 4 metric for baseline and gradual forced attention arch.....73
Table 52: METEOR metric for baseline and gradual forced attention arch. 73
Table 53: ROGUE L metric for baseline and gradual forced attention arch. 73
Table 54: CIDer metric for baseline and gradual forced attention arch. 74

LIST OF FIGURES

Figure 1: Overall architecture of the neural network. 40
Figure 2: Encoder network - Convolutional Neural Network. 41
Figure 3: Attention Mechanism 42
Figure 4: LSTM cells - a set of these comprise the decoder network. 43
Figure 5: Randomly sampled image from clock dataset (MSCOCO (Lin at al., 2014)) 64
Figure 6: Randomly sampled image from clock dataset (MSCOCO (Lin at al., 2014)) 65
Figure 7: Randomly sampled image from train dataset (MSCOCO (Lin at al., 2014)) 70
Figure 8: Randomly sampled image from train dataset (MSCOCO (Lin at al., 2014)) 70
Figure 9: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014)) 74
Figure 10: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014)) 75
Figure 11: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014)) 76
Figure 12: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014)) 77

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory Cell
GRU	Gated Recurrent Unit
MSCOCO	Microsoft Common Objects in Context

GLOSSARY

Artificial Neural Networks: They are densely interconnected adaptive processing elements capable of performing massive parallel computations (Basheer & Hajmeer, 2000). The neurons are based on the neural structure of the brain. They process information in a similar way the human brain does. (Maind & Wankar, 2014).

Attention: Refers to the process by which organisms select a subset of available information upon which to focus for enhanced processing (often in a signal-to-noise-ratio sense) and integration. (Ward, 2008).

Deep Neural Networks: Neural Networks composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. (LeCun, Bengio & Hinton, 2015).

ABSTRACT

Author: Devarapalli, Hemanth. MS
Institution: Purdue University
Degree Received: December 2018
Title: Forced Attention for Image Captioning.
Major Professor: Dr. Julia Rayz

Automatic generation of captions for a given image is an active research area in Artificial Intelligence. The architectures have evolved from using metadata of the images on which classical machine learning was employed to neural networks. Two different styles of architectures evolved in the neural network space for image captioning: Encoder-Attention-Decoder architecture, and the transformer architecture. This study is an attempt to modify the attention to allow any object to be specified. An archetypical Encoder-Attention-Decoder architecture (Show, Attend, and Tell (Xu et al., 2015)) is employed as a baseline for this study, and a modification of the Show, Attend, and Tell architecture is proposed. Both the architectures are evaluated on the MSCOCO (Lin et al., 2014) dataset, and seven metrics: BLEU – 1, 2, 3, 4 (Papineni, Roukos, Ward & Zhu, 2002), METEOR (Banerjee & Lavie, 2005), ROGUE L (Lin, 2004), and CIDer (Vedantam, Lawrence & Parikh, 2015) are calculated. Finally, the statistical significance of the results is evaluated by performing paired t tests.

CHAPTER 1. INTRODUCTION

Image captioning is the generation of properly formed English sentences which describe an image (Vinyals, Toshev, Bengio & Erhan, 2015). It is a complex task involving multiple individual parts. The algorithm has to read and understand the image, and then express its comprehension in the form of text. The text has to be legible, and make sense. Any attempt to tackle this problem would have to involve image analysis, and text generation.

Image captioning systems typically consist of two independently functioning parts – an Image analysis part, and a Text generation part. Both the Image analysis part and Text generation part are different configurations of Artificial Neural Networks. Basheer and Hajmeer (2000) define Artificial Neural Networks as computational structures comprised of interconnected simple processing elements capable of performing parallel computations. The Image analysis part is a Convolutional Neural Network, which is a specialized Artificial Neural Network architecture designed to work with two dimensional data. Convolutional Neural Networks are able to handle the invariances of two-dimensional shapes with their local connection patterns and constraints on their weights (LeCun, Bottou, Bengio & Haffner, 1998). For the Text generation part, Recurrent Neural Networks are used. The Recurrent Neural Networks are a class of specialized Artificial Neural Networks designed for sequences, and have a hidden state which, in the case of Text generation, is used along with the input to predict the next word (Sutskever, Martens & Hinton, 2011). Additionally, Attention has been used in Image Captioning networks to further improve the performance by using feature maps of objects instead of lossy fixed length vector.

Attention mechanism is seen in many sequence models (Vaswani et al., 2017). Prior to attention, neural networks used to encode the entirety of the image into a fixed length vector, which is used to generate captions. The neural networks did not try to concentrate on specific parts of the image, which would have higher significance for the caption. This could lead the neural network to generate vague captions, ignoring the main subject of the image. Attention mechanisms are used to rectify this problem. These neural networks generate caption which contain the object which was attended to. However, the parts of the image to attend to is left to the neural network. This

results in a variety of captions being generated, each concentrating on different set of objects in the image.

1.1 Statement of the Problem

Attention based Image Captioning neural networks attend to a few objects in the image, and generate a caption. However, there is no guarantee that all objects will be attended to, while captioning. This study attempts to design a network that can attend to the object, generating a caption which talks about the object. Hence, generated caption is able to give context to the object's presence in the image.

1.2 Research Question

The research question addressed in this thesis is:

- How much does the performance of an Image Captioning Network improve with the addition of forced attention?

The question will be answered by accomplishing the following tasks:

- Pick objects to focus on, and prepare the dataset.
- Establish baseline performance using the Soft Attention Model of the Show, Attend and Tell architecture (Xu et al., 2015).
- Implement forced attention on the Show, Attend and Tell architecture (Xu et al., 2015).
- Evaluate the performance of the forced attention network and compare it to the baselines.

1.3 Scope

The aim of the study is to add the ability to concentrate on a specific object to an image captioning neural network. This would involve finding a way to specify a common element (*hereby referred to as focus object*) in the dataset, on which the neural network will concentrate. The neural network then attempts attend to the focus object while generating the captions. For the neural network, existing implementations of encoding and decoding networks used in Image Captioning networks are reused. The majority of the study will be focusing on adding the ability to specify the focus object, and make the neural network attend to it. The performance of the network is evaluated using the same metrics which are used in the baseline (Xu et al., 2015).

1.4 Significance

Image captioning played an important part in the field of Artificial Intelligence. Being able to caption an image accurately finds use in a wide variety of areas. Images can be quickly categorized, and tagged – which helps with image search engines. Identifying the parts of an image in a person's library helps with search, which was traditionally based only on dates and locations. This is seen in Google's and Microsoft's cloud offerings, where the uploaded pictures are tagged according to the user, their friends, animals and environments.

The ability to generate captions, which include context for the focus object, would help improve image tagging and categorization. Search engines would be able to filter results with even more precision, thanks to the neural network's ability to concentrate on specific objects.

Sentiment analysis of pictures is another research area where focused image captioning finds use. Accurate captions help identify the mood of the scene, as well as the associated objects and people. This can help in gauging the sentiment not just for the whole image, but narrowed down to specific objects and people. Additionally, having the textual representation helps with tagging and indexing, so that they can be retrieved and used later.

The sentiment analysis use-case ties into advertising and merchandising. Brands can accurately identify how their merchandise is being shown in pictures shared on social media. It allows a company to understand how or what scenarios the object usually appears in, and thereby change or adapt their ad-campaigns, or add new merchandise.

Focused image captioning can also be used to generate massive amounts of training data for other machine learning algorithms and neural networks. Having an appropriately captioned image dataset is extremely time consuming for humans to create. However, having a neural network enables researchers to generate massive amounts of training and validation data. As new datasets are hard to come by, focused image captioning can be helpful for this use case.

Focused image captioning can be applied on a video as well, where the object can be tracked, and all the actions performed can be captioned. Captioning video itself is another active research area in Artificial Intelligence. The approaches taken can be broadly split into two. First, where the video is split into its individual frames, after which each image is processed separately. The second also

involves splitting the video into separate frames. However, the individual images aren't processed separately, rather, they're fed together in batches to preserve the temporal relations. Focused image captioning can be adapted to either of the two methods, to get better results.

Finally, the ability to focus on an object while captioning in a video is useful in video surveillance. Apart from tracking a particular subject, the constant captioning of the subject can help build index-able and searchable history, ready to be retrieved and used for further analysis.

1.5 Assumptions

Assumptions for this study are primarily related to datasets. The following assumptions were made for this study:

- The captions for the images in MSCOCO dataset are assumed to be the ground truth. MSCOCO was the largest image captioning dataset available during the time of this study.
- MSCOCO is assumed to be diverse enough for the trained model to generalize over other/newer datasets. This is a common occurrence in machine learning, where the model is not able to generalize while running on other/newer datasets.
- The performance metrics used for evaluation are assumed to accurately convey the quality of the generated caption.

1.6 Limitations

The study is undertaken with the following limitations:

- The metrics obtained on the pruned dataset by previous models is used as the baseline for evaluations.
- The focus objects must be from the MSCOCO dataset. As the network is trained on the MSCOCO dataset, it is able to identify and caption objects only from this dataset.
- The attention forcing mechanism is employed for soft attention.

1.7 Delimitations

The study acknowledges the following delimitations:

- There are more attention mechanisms available, like hard attention. However, they are not being worked upon for this study. Soft attention is used because it is easier to modify. The attention maps from soft attention models have larger gradients, and hence forcing attention is simpler.
- Given the fast pace of improvements in Convolutional Neural Networks, there are many different configurations of networks available for Image Captioning. Though they all have a similar overall structure, the individual components are quite different. This study uses residual networks, specifically Resnet50.
- There may be newer, larger captioning datasets available in the future, however, this study is restricted to use the MSCOCO dataset.

CHAPTER 2. LITERATURE REVIEW

This chapter is a summary of the recent research literature in Image Captioning, along with its constituent tasks - Image Classification and Text Generation.

2.1 Introduction

The ability to caption an image is a fundamental problem in Artificial Intelligence. This particular problem ties together both Natural Language Processing and computer vision (Vinayals, Toshev, Bengio & Erhan, 2015). Hence, Image Captioning requires effort in both those fields for implementing a solution. There have been many attempts in this direction; and the past decade has seen tremendous progress.

First, early attempts in image captioning before the prevalence of deep neural networks are described. The advent of huge datasets and cheap computing power enabled the resurgence of deep neural networks. Hence, work done on creating new datasets is covered next.

As image captioning consists of text generation and image analysis, there is ample work done in either field which has to be taken into consideration. Hence, next in line are deep learning methods for text generation and image analysis. After which, work done in combining both fields to generate descriptions of images are reviewed.

Then, attention, which was another pivotal mechanism in deep learning is described. Finally, the evaluation metrics used for this study are reviewed.

2.2 Early Attempts

Early image captioning involved assigning keywords, or tags to a given image. Pan, Yang, Faloutsos and Duygulu (2004) proposed a graph-based approach (GCap) for this particular problem. The training dataset had images, with contiguous blobs or regions having a corresponding keyword. Then, a graph is built with nodes for images, keywords and contiguous regions. Any query image is segmented into contiguous regions, for which similar nodes are retrieved from the

graph. The created graph is traversed from the nearest node to generate the keywords. The authors used Corel image dataset of 630Mbytes and calculate the percentage of correct keywords. GCap consistently outperformed the previous work by Duygulu, Barnard, de Freitas, and Forsyth (2002).

Aker and Gaizauskas (2010) attempted to generate image descriptions by using the image's metadata – the place names and tags associated with the image. The authors argued that just using the GPS coordinates attached to the image doesn't yield enough information for captioning. Aker & Gaizauskas (2010) derived an n-gram language and dependency pattern models using their earlier works. The method the authors developed applied only to images with static features – either man made or natural (Aker and Gaizauskas, 2010). The authors used a bi-gram language model for each object type corpus. The dependency pattern was derived by using the Stanford parser (Klein & Manning, 2003). The approach involved querying Yahoo! Search engine with the image's toponym. The top 30 results were parsed using an HTML parser to extract the text, and the text was sent to the summarizer. While building the summarizer, Aker and Gaizauskas excluded any Virtual Tourist sites, as those websites were used as a part of the training corpus itself. The summarizer was an extractive, query-based multi-document summarization system. It took two inputs: a toponym for the image, and a set of documents to be used for generating the description. As the summarizer may generate multiple sentences, each sentence is scored. For scoring the sentences, linear function with weighted features was used, where the weights were learned using linear regression. Images and their respective descriptions from Virtual Tourist website were used for training the linear regression model. Finally, Aker and Gaizauskas used ROUGE (Lin, 2004) and manual readability as metrics for evaluating their results. Their model improved over their previous results, improving ROGUE R2 from 0.095 to 0.102, and ROGUE RSU4 from 0.145 to 0.155. However, when evaluating manual readability compared to Wikipedia, the authors mentioned that their model performed better in one feature – grammar, but still needed improvement in the remaining – clarity, focus, coherence, and redundancy (Aker & Gaizauskas, 2010).

2.3 Advent of new Datasets

Ordonez, Kulkarni and Berg (2011) introduced a new method of captioning images, along with a new dataset with a million images. Their new dataset contained images, along with the associated

captions written by people. Along with the dataset, the authors proposed two methods to generate captions. First was a description generation method which utilizes global image representations to retrieve and transfer captions from a dataset to the query image. The second method involved utilizing both global representations and direct estimates to produce relevant image descriptions.

To build the dataset, Ordonez et al. (2011) queried Flickr with pairs of query terms, generating a large but noisy set of photographs and their associated text. Then, this noisy set of pictures and text was filtered, so that the descriptions attached to each photograph are relevant, and visually descriptive. For an input query image, global similarity with the dataset is computed. After finding the closest matching candidates, the authors transfer the descriptions. The image's constituent Objects, Stuff, People, Scenes and Frequency measure (TFIDF) were used to rank the captions. Finally, by training a linear regression model with the generated caption and its resulting BLEU score, the authors predicted the best caption. For the second method, 100 most similar images to the query were selected. From these images, the constituent Object, Stuff, People, Scene and Frequency ranks are extracted, on which an SVM is trained with 5-fold cross validation. The authors mention seeing some reasonable results – sometimes describing a scene extremely well, some even having good description of attributes, or "being poetic" (Ordonez, Kulkarni & Berg, 2011). However, there are equally irrelevant descriptions generated too. Overall, the model achieved a BLEU-1 score of 0.125 when used with the linear SVM. The authors switched the linear SVM with a linear regression model, which resulted in a BLEU-1 score of 0.121.

Microsoft (Lin et al., 2014) released the MSCOCO dataset, also called common objects in context, in 2014. Lin et al. (2014) gathered images of complex everyday scenes – which contain common objects in their natural context. The dataset is labeled using per-instance segmentations – allowing precise object localization. This new dataset contained 328k images with 2.5 million labeled object instances. These images were further categorized into 91 different types of objects. The authors used Amazon's Mechanical Turk (Amazon Mechanical Turk, 2018) to add the features to the dataset. The dataset had to be annotated to add the category labels, presence of an object, and its localization. All of these were delegated to the Mechanical Turk annotators, which required 70,000 worker hours.

Google (Sharma, Ding, Goodman & Soricut, 2018) published a massive dataset containing ~3.3 million image and caption pairs called Contextual Captions. The authors created this dataset by processing billions of webpages in parallel, from which the candidate image and caption were extracted. Only the images with a minimum height and width of 400px were used. Each image's alt-text was extracted and processed by Google Cloud Natural Language API, focusing on the Parts of Speech, sentiment/polarity and profanity annotations (Sharma, Ding, Goodman & Soricut, 2018). Similarly, Google Cloud Vision service was employed to assign class labels to the images. These class labels are used along with the extracted alt-text to create candidate captions. This involved removal of noun modifiers, dates, durations. The identified named-entities were appropriately substituted. In addition to publishing this massive dataset, the authors trained two image main captioning models, one being a model similar to the baseline used for this study – Show and Tell (Vinyals et al., 2015), and the second - a pure attention model (Vaswani et al., 2017). The authors report that both models when trained on this new dataset perform better than the same models trained on MSCOCO.

2.4 Deep Learning Techniques

The availability of large datasets like MSCOCO along with cheaper computing infrastructure helped usher in the era of deep learning (Jones, 2014). The Image captioning networks can be split into two parts – the network which analyzes the input image, and the network which generates the caption. Both these neural networks had their own staple methods and properties, along with their own breakthroughs.

2.4.1 Usage of Deep Learning Techniques for Text Tasks

Though there are many deep learning based approaches in this direction, this review concentrates on few pivotal works – use of Recurrent Neural Networks (RNN), namely, Long Short-Term Memory units (LSTM); Convolutional Neural Networks (CNN) and Visual Attention. All of these built upon their previous work and advanced the state-of-the-art further, improving the performance.

RNNs themselves have been successfully used to generate sequences in various domains – like music (Boulanger-Lewandowski, Bengio & Vincent, 2012) and text (Sutskever, Vinyals & Le,

2014). They are fuzzy, and hence do not have exact templates to match the predictions. Fuzzy predictions also have the benefit of not suffering the curse of dimensionality (Graves, 2013). Any arbitrary sequence can be modelled by a large RNN. However, in practice, RNNs have amnesia, where the prediction is based only on the previous few inputs. This means they cannot recover from past mistakes, as they will keep continuing to do the same. Furthermore, generic RNNs also suffer from vanishing (or exploding) gradients – where the error either vanishes or explodes if the time steps are long. LSTM cells (Hochreiter & Schmidhuber, 1997) were designed to alleviate this problem.

Long Short-term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) units are Recurrent Neural Networks (RNNs) with architecture designed to be better at storing and accessing information than generic RNNs (Graves, 2013). The LSTM cells have a hidden state, which stores information about the previously generated outputs. The information about previous outputs is used while generation of each word. Sundermeyer, Schlüter and Ney (2012) applied LSTM cells to language modelling tasks on English and French datasets. Penn Treebank and Quaero French datasets were used for their study. By using LSTMs along with standard neural networks, the perplexity was improved by 8%.

Bahdanau, Cho and Bengio (2014) proposed a language translation mechanism using only RNNs. They built an encoder-decoder network which could be jointly trained. Previous efforts involved translating the source sentence to a fixed length vector (Bahdanau, Cho & Bengio, 2014). This meant the algorithm had to compress all the information from the arbitrary length input sentence into a fixed length vector – which is lossy. Instead, the proposed neural network encodes the input into a sequence of vectors, and chooses a subset of these vectors while decoding. This neural network has a Bidirectional RNN as an encoder, whose output is given to the decoder which emulates searching through a source sentence. The encoder has forward and backward RNNs each having 1000 hidden units, whereas the decoder network has 1000 hidden units. Both segments use a multi-layer network with a single maxout hidden layer to compute the conditional probability of each target word. The authors (Bahdanau, Cho & Bengio, 2014) report that their neural network architecture scores 21.5 and 26.75 compared to the previous architecture's scores of 13.93 and 17.82 on the BLEU metric.

Few of the commonly used regularization mechanisms (Hinton et al., 2012) in generic deep neural networks could not be adapted for RNNs, due to their recurrent connections (Zaremba, Sutskever & Vinyals, 2014). Dropout was one such mechanism, where neurons are randomly turned off was often used to avoid overfitting. Large and complex RNNs often overfit, which made generalization a problem. Therefore, practical applications of RNNs had smaller models to avoid overfitting. Zaremba, Sutskever and Vinyals (2014) presented a simple recipe for applying dropout for LSTMs, which successfully reduces overfitting. The main contribution of their paper was the application of dropout only to the non-recurrent connections.

2.4.2 Deep Learning Techniques for Image analysis and Classification

The field of deep learning-based Image analysis and Classification has seen extensive work being done. There are many popular networks (for example, ResNet (He, Zhang, Ren & Sun, 2016), VGG16 (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky, Sutskever & Hinton, 2012)) that give good results on image datasets. They all use the Convolutional Neural Network, in which a set of kernels is used to convolve the input image and generate a feature map. Over the past few years, there have been quite a few novel uses of Convolutional Neural Networks for Image Classification, where various new techniques like multi-scale, sliding window (Sermanet, Eigen, Zhang, Mathieu, Fergus & Lecun, 2013) and very deep convolutional networks (Simonyan & Zisserman, 2014) were used.

Overfeat architecture was proposed by Sermanet, Eigen, Zhang, Mathieu, Fergus and Lecun (2013). Here, the authors show that by making the network perform multiple tasks, the resulting accuracy is boosted. They use sliding windows over multiple-scales for their network. Then the network is trained to classify the central pixel of the viewing window, using the window contents as the context. The network is given three main tasks – classification, localization, and detection. Each task can be considered as the subtask of the next. For the classification, the network uses a 7-layer neural network, of which five are Convolutional Neural Networks, and two are fully connected networks. This 7-layer neural network uses a sliding window along with multi-scale classification. Next is the localization task, where the network's classification output is sent to a regression layer, which generates possible bounding boxes. Hundreds of bounding boxes are

generated, which are merged to get the best possible box. The detection task is at the end, which is similar to the classification task. The only difference being that its necessary to predict a background class even if there is no object present. The Overfeat architecture (Sermanet, Eigen, Zhang, Mathieu, Fergus & Lecun, 2013) won the ILSVRC12 and 13 competitions with 29.9% error.

Donahue et al. (2014, January) proposed a way of using pre-trained networks for smaller datasets. Deep neural networks typically require large amounts of data (Sze, Chen, Yang & Emer, 2017). So, datasets which may not have enough labeled data cannot be used for training a deep neural network. Donahue et al. (2014, January) propose a supervised pre-training of the network – on a large labeled dataset. The network is able to learn salient features by themselves without the need for hand-engineering the features. Then, the authors pick the 6th and 7th layers of the network – fully connected layers which come after the Convolutional layers. These layers are selected to capture the feature representation that is generic enough to be used with a smaller dataset, while not being noisy. Then, an SVM is trained using the feature representations and performance is evaluated. The authors mention that the 6th layer performs much better for the Object recognition and domain adaptation tasks. However, for subcategory classification tasks and scene recognition tasks, the 7th layer is reported to perform better. Donahue et al. (2014, January) mention that their architecture dramatically outperforms the baselines.

Simonyan and Zisserman (2014) investigated the effect of depth of the Convolutional Neural Network on accuracy. They used smaller 3x3 filters instead of large filters, and increased the depth to get similar accuracy as the large filters. This is in contrast with other top performing models of ILSVRC-12 and 13 (Krizhevsky, Sutskever & Hinton, 2012) – which used large filters in the initial layers itself. These models had filters as large as 7x7 and 11x11 (Krizhevsky, Sutskever & Hinton, 2012), which are in stark contrast with the approach taken in Simonyan and Zisserman (2014). Instead, the depth was increased, going as deep as 19 layers which improves the accuracy. Simonyan and Zisserman (2014) ran their models on the ILSVRC-12 dataset (Russakovsky et at., 2015), which had 1.3M training images with 1000 classes. The model was named VGG (after the lab where it was created, Oxford's Visual Geometry Group), which outperformed the previous generation best performers. The authors secured the 2nd place in the ILSVRC-14 challenge, with

7.3% test error. Their model is competitive with the GoogLeNet (Szegedy et al., 2015). The authors claim this is remarkable, as they used just combined two models as an ensemble, which is significantly less than what is used in most of the ILSVRC submissions. This shows that increasing the filters isn't always required for complex and large volume tasks.

2.4.3 Image Captioning with Deep Learning

The previously mentioned efforts in both text generation and image analysis and classification are tied together for image captioning. The archetypal approach seen in Image Captioning architectures is a Convolutional Neural Network for the image analysis (encoder), and then a Recurrent Neural Network for the caption generation (decoder). Numerous innovations have been done in each segment to improve the performance.

Vinyals, Toshev, Bengio and Erhan (2015) presented a new neural network architecture called Show and Tell, which tries to maximize the likelihood of the target description sentence given the image. Show and Tell uses a Convolutional Neural Network to detect objects within an image. Then, a Recurrent Neural Network takes these detected objects and generates the caption. For the Recurrent Neural Network, they use the Long Short-Term Memory units (LSTMs) (Hochreiter & Schmidhuber, 1997).

For the Show and Tell neural network architecture, the authors used BLEU metric (Papineni, Roukos, Ward & Zhu, 2002) , which is the most commonly used metric for evaluating image description. It is a form of precision of word n-grams between generated and reference sentences. The second metric was manual work, for which Amazon's mechanical turk was employed. The dataset used for the study was MSCOCO, which was the largest dataset for this task at the time. Show and Tell achieved near to the state of the art, and was competitive to the human performance in all categories. This neural network architecture achieved a BLEU-4 score of 27.2. Many authors would later build on this work and improve the scores further.

Fang et al. (2015) proposed a unique neural network architecture, where they trained visual detectors for words that commonly occurred in captions. The authors used multiple instance training to train visual detectors for words, whose outputs are used as conditional inputs for the

max-entropy language model. This is in contrast with the usual approach taken in this field – using Convolutional Neural Networks to map images to vectors, after which the vectors are fed to a Recurrent Neural Network to generate a caption. Another approach is using pure statistical approaches, for guiding language models using images. The authors, instead, use both, neural networks and statistical approaches . Starting with Convolutional Neural Networks, which detect object regions on images. Then, the regions are associated with words using multiple instance learning. The authors try to minimize the a priori assumptions when looking at how sentences should be structured. Finally, text generation problem is solved as optimization problem along with a search for the most likely sentence. Therefore, the visual detector architecture (Fang et al., 2015) learns to extract nouns, verbs and adjectives from regions in the image. The extracted words are used to guide the language model to which generates the text. This neural network architecture became the state of the art at the time, producing a BLEU-4 score of 29.1% on the MSCOCO dataset. The authors mentioned that their architecture outperformed the previous models on all 14 metrics of the MSCOCO image captioning task, and managed perform equally or better than humans on 12 of the 14.

Chen and Lawrence (2015) explored another facet of Image Captioning. They looked at the bi-directional mapping between the images and the description. The aim was to learn this mapping using a Recurrent Neural Network. This approach is unique, as most previous approaches map both the image and the sentence to a common embedding space. The bi-directional nature meant that their model could generate images from sentence features, and sentences from image features. The Bi-Directional neural network architecture used an existing pre-trained model for classifying the image, and then a Recurrent Neural Network to generate the description. The authors added a recurrent visual hidden layer, which attempts to reconstruct the visual features from the previously generated words. So, this neural network architecture can compare its visual memory of what it already said, to what it currently observes. Using this knowledge, the neural network architecture predicts what to say next. The authors mentioned that the language model contained anywhere from 3000 to 20000 words. A maximum entropy model was used to reduce the perplexity, which was learnt from the training corpus. The authors (Chen & Lawrence, 2015) report that their neural network architecture consistently beat the Midge and Babytalk (Kulkarni et al., 2013) models, and performs near to humans when looking at the BLEU scores on MSCOCO dataset. The authors

reported that they achieved a BLEU, METEOR and CIDer scores of 18.4, 19.5 and 53.1 on the MSCOCO dataset, where human performance was at 21.7, 25.2 and 85.4 respectively.

Karpathy and Fei-Fei (2015) published a paper called Visual-Semantic Alignment. They used Convolutional Neural Networks over image regions, Bi-Directional Recurrent Neural Networks for the sentences, and aligned both of them using a multimodal embedding. Their trained network uses this inferred alignment to generate novel descriptions of a given image.
According to the authors, the two main contributions were – creating a new model which can infer the alignment of the segment of the sentence and the region of the image, and then creating a network which was able to get state of the art results on Image-Sentence ranking tasks on Flickr8K, Flickr 30K. The Visual-Semantic Alignment architecture outperforms the retrieval baselines on full images as well as a new dataset of region-level annotations. It also outperformed the previous work done in this task – getting a R@1 (Recall @ K, where K can be 1, 5, 10) score of 22.2 on Flickr 30K compared to a previous best of 18.4. Karpathy and Fei-Fei (2015) also talk about the limitations in their model, as their neural network architecture is restricted by the input image resolution. The authors mention that approaches using multiple saccades around the image to identify all entities, their mutual interactions, and wider context before generating the description would help.

2.5 Further Innovations in Image Captioning

All the novel approaches described in the previous sections were all reused, improved upon, and assembled in even better neural network architectures.

SentiCap (Mathews et al., 2016) is a neural network architecture with the aim to generate captions with sentiment – they appear more human-like, and contain more descriptive adjectives. Here, the authors employ two CNN-RNN networks. Each of the neural networks is an archetype of convolutional image encoder - recurrent text decoder neural network architecture. One is for generating the factual image description, and the other is specifically for generating words with sentiment. SentiCap has an additional switching gate, which is employed to switch and combine the outputs. This switching gate generates a probability of switching between the two RNNs at each time, with a single layer network taking the hidden states of both RNNs as input (Mathews

et al., 2016). Another point to note was that the authors trained the networks for positive and negative sentiments separately, as both could be valid of majority of the images. For evaluating SentiCap, the authors employ both automatic metrics as well as crowd-sourced judgments with Amazon Mechanical Turks (AMTs). The automatic metrics are the BLEU, ROGUE, METEOR, and CIDER metrics. Whereas, the crown-sourced metric involved AMT tasks, where each image had two captions - one generated from the factual description generator and one with sentiment. The AMT workers had to rate the descriptiveness, and pick the more positive or negative caption. To ensure the quality of manual evaluation, the authors made sure at least two people agreed on the positive or negative sentiment selection. The authors reported that their network had significantly more sentences with sentiment words than any of the three baseline methods they picked. They also note that, on average, their network was judged by the AMT workers to have stronger sentiment compared to the baselines. Finally, SentiCap is reported to generate 95.7% novel captions, compared to the author's factual caption network, which generated 38.2% novel captions - on MSCOCO (Mathews et al., 2016).

Another interesting use-case which finds home in this field is question answering systems. As the neural networks are able to comprehend the content of an image, and generate its textual representation, work has been done to take this further to build question answering systems.

In this particular study (Ma, Lu & Li, 2016), the authors emphasize on the network's ability to learn the inter-modal interactions. Their neural network architecture is jointly trained to produce the answer for a given image and its question. The unique component of the paper is that instead of the usual approach of using RNNs for the text generation, the authors use CNNs. This contrast is explained by the authors, where they compare their CNN only neural network architecture with the CNN-RNN architecture. The authors explain that such networks ignore the different characteristics of the questions and answers. The questions - being lengthy, and have a somewhat similar structure, differ from answers - which are usually short, and tend to be a single word. Yet another justification is offered by the authors for their use of CNNs for text generation. The usual approach is using LSTM cells to jointly model the image and question by treating the image as an independent word, and appending it to the question at the beginning or the end (Ma, Lu & Li, 2016). The authors however argue that treating the image as a word cannot effectively exploit the

relations between the image and the associated question. The authors propost their neural network architecture, which consists of three individual CNNs. The image encoder is a CNN, and then a sentence CNN for generating the question representation, and finally a multimodal CNN, which fuses the image and question representation to create the joint representation. This is finally fed to a softmax layer to generate the answer. This whole neural network is jointly trained in an end-to-end fashion. The metrics used for evaluating this network is the Wu-Palmer similarity (Wu & Palmer, 1994). Also called WUPS, it calculates the similarity between two words based on their common sub-sequence in a taxonomy tree (Ma, Lu & Li, 2016). WUPS requires threshold parameters, which are set to 0.0 and 0.9 for the WUPS@0.0 and WUPS@0.9 respectively. Apart from WUPS, the authors use the accuracy, measuring the proportion of the correctly answered questions. The proposed neural network architecture (Ma, Lu & Li, 2016) outperforms all the compared models/networks on the DAQUAR-All dataset (Malinowski & Fritz, 2014). Similarly, on the DAQUAR-reduced dataset, their network again outperforms the competition. Finally, on the COCO-QA dataset (Ren, Kiros & Zemel, 2015), the proposed neural network architecture (Ma, Lu & Li, 2016) outperforms all the competing networks.

Quite recently, there was work done in extracting facts from Images. Sherlock (Elhoseiny et al., 2017) is one such neural network architecture consisting of two components for encoding – a CNN for the visual processing, and a word2vec for the associated caption. The Sherlock architecture then attempts to associate the caption and image to generate facts. To do so, the authors (Elhoseiny et al., 2017) attempt to minimize the distance between the two embedding spaces – the feature map, and the word embedding. The Sherlock architecture is reported to understand various objects, actions, as well as interactions between objects. The authors also create a large-scale benchmark for such tasks - containing over 814,000 examples and 202,000 unique facts, and show the value of relating facts by structure using the proposed model (Elhoseiny et al., 2017). The Sherlock architecture is reported to get the best mean Average Precision over all competing models.

Aditya (2017) worked on building explainable image understanding neural networks, which can be used to generate captions and answer questions. This work built upon previous work in that direction (Aditya et al., 2015)– based on visual common sense, and scene description graphs. The author proposes a neural network architecture called DeepIU (Aditya et al., 2016), where the visual

data is combined with background knowledge. It is then looped through visual and reasoning modules until the results are exhausted, and is able to answer questions about the image, or generate description of the said image. This particular architecture takes inspiration from how humans function – where the knowledge is continuously refined by asking questions. Another noteworthy task tackled by the author is the ability to answer riddles. The author reports that the proposed network is able to achieve few interesting on riddles which are harder for humans as well. DeepIU (Aditya et al., 2016) performs better than (Karpathy & Fei-Fei, 2015) while measuring recall.

Reference based LSTM (Chen et al., 2017) is a simpler neural network architecture which uses the classic CNN-RNN architecture. However, the authors employ weighted training for the neural network. During training, the authors assign different weights to words. Therefore, the neural network is able to assign importance, and hence can better learn key information required for captioning. Higher weights were assigned to words which indicate important elements, such as the subject, etc. While generating the caption, a consensus score is utilized to exploit the reference information of neighbor images (Chen, Ding, Zhao, Chen, Liu & Han, 2017). Reference based LSTM (Chen et al., 2017) uses the VGG-16 (Simonyan & Zisserman, 2014) as the CNN encoder for extracting image features. For the RNN decoder segment, LSTM cells are used. The authors compared their proposed neural network architecture with sever state-of-the-art networks. The proposed approach performs better on all metrics (BLEU-1, 2, 3, 4, METEOR, CIDER and ROGUE) compared to the previous state-of-the-art neural networks.

2.6 Advent of Attention Models

In CNN-RNN architectures, the algorithm is left on its own to find out the entities and generate sentences. However, there can be many instances where the algorithm does not look at the relevant parts of the image or the algorithm does not pay enough attention to the relevant parts of the image. This happens because the neural network compresses the features from the input image into a fixed length vector. Attention models attempt to rectify that problem, and allow the model to attend to specific parts of the image. With Attention, the neural network can use the features detected in the image separately, bypassing the lossy fixed length vector representation.

Ba, Mnih and Kavukcuoglu (2014) presented a new approach – using attention for generating descriptions of images. The authors employed reinforcement learning, which is usually used in training autonomous agents. The proposed neural network was trained to attend to specific parts of the image using reinforcement learning, and hence, the neural network could learn to both localize and recognize multiple objects. The authors (Ba, Mnih & Kavukcuoglu, 2014) take inspiration from the way humans perform – continually moving the fovea to the next relevant object, recognizing it, and adding it to the internal representation of the sequence. The authors perform a multi-resolution crop of an input image, called a glimpse (Ba, Mnih & Kavukcuoglu, 2014). The proposed neural network architecture uses each glimpse to update its internal representation, and then outputs the next glimpse location as well as the next object in the sequence. The authors let the model continue until it decides there are no more objects left to process. As mentioned before, this lets the neural network to do both localization and recognition of multiple objects. The authors evaluated their neural network architecture on the standard SVHN sequence recognition task (Netzer et al., 2011), where it outperforms the state-of-the-art. The proposed neural network architecture got a test error percentage of 3.9%, compared to 3.96% of the state-of-the-art. Another unique aspect is that their model uses less parameters and is less computationally intensive than the CNNs which look over the entire image, highlighting the fact that attention mechanisms can improve the accuracy and efficiency of CNNs.

Building on Show and Tell (Vinyals, Toshev, Bengio & Erhan, 2015) neural network architecture, Xu et al. (2015) introduced a new model – Show, Attend and Tell. Xu et al. (2015) show visually how their proposed neural network architecture is able to automatically learn to fix its gaze on relevant and salient objects while generating the corresponding words in the output sequence. The authors introduced two attention mechanisms – a "soft" deterministic mechanism which is easily trainable by standard back propagation methods, and a "hard" stochastic attention mechanism which is trainable by maximizing an approximate variational-lowerbound. Apart from this, the authors were able to visualize this model, and hence could see "where" and "what" the model was paying attention to. Finally, Show, Attend and Tell (Xu at al., 2015) achieved the state-of-the-art performance on the three popular benchmark datasets – Flickr8K (Hodosh, Young & Hockenmaier, 2013), Flickr30K (Young at al., 2014) and MSCOCO (Lin et al., 2014). Show, Attend and Tell (Xu at al., 2015) scored BLEU-1 score of 67 on Flickr 8K with both hard and soft attention whereas

the previous best were 63 and 65.6. In the Flickr 30K, the BLEU-1 scores for the proposed soft and hard attention neural network architectures were 66.7, and 66.9. The previous best was 66.3. Similarly, in MSCOCO, the proposed soft and hard attention neural network architectures got BLEU-1 scores of 70.7 and 71.8 compared to the previous best – 70.8.

Lu, Xiong, Parikh and Socher (2016) proposed another novel approach using attention. They proposed the use of a "visual sentinel", which the decoder network can use when generating non-visual words. This is because the decoder doesn't require any visual cues while generating non-visual words like "on", "of". So, the neural network decides whether to attend to the image, and where to attend, while extracting meaningful information. If the visual cues aren't required, it uses the sentinel instead. This visual sentinel is a new long short-term memory (LSTM) extension, which provides the fall back option for the decoder. Hence, the authors (Lu, Xiong, Parikh & Socher, 2016) a new encoder-decoder framework that can decide automatically when to look at the image, and when to look at the language model (sentinel) while generating the next word. The authors proposed a newer attention model and built upon it for making the visual sentinel. Their proposed neural network architecture improves the BLEU 1, 2, 3, 4, METEOR and CIDer scores on Flickr30k and MSCOCO datasets compared to previous architectures.

Further improvements to attention mechanism was the use of Global-Local attention (Li et al., 2017). Here, the authors propose the use of two image encoding networks – one for extracting global features, and one for local features. Then, global features are fused with the local features using an attention mechanism. This is used by the decoder network to generate the captions for an image. Global-Local attention architecture (Li et al., 2017) uses VGG-16 (Simonyan & Zisserman, 2014) for the global feature extraction, and a Faster R-CNN (Ren, He, Girshick & Sun, 2015) for the local features. The proposed attention mechanism then dynamically weights each feature along with the sentence generation procedure. The decoding layer employed a two-layer LSTM, which generates the caption for the input image. The authors report state-of-the-art performance on the MSCOCO dataset.

The task of generating a narrative, given a set of related images, was tackled by Let Your Photos Talk (Liu, Fu, Mei, & Chen, 2017). For this task, the network has to remember the previously read

photos, as well as the generated text while generating the words in a sentence. Here, the authors use Bi-directional attention based RNNs (BARNN). They argue that a cross-modal embedding model can handle the inherent visual variance in a stream of pictures, hence can represent the underlying story. The authors propose a BARNN framework which includes their new kind of GRU called skip-GRU (Liu, Fu, Mei & Chen, 2017). It is able to handle the implicit semantic relations, so as to enforce coherence in the generated sentences. The proposed neural network architecture has a CNN which was extracted from the popular VGG-16 network (Simonyan & Zisserman, 2014). It is followed by a classic feed forward/fully connected network. The output of this feed forward network is fed to the BARNN to generate the corresponding sentence vectors. The outputs of both the feed forward network and BARNN are then matched in an embedding space, from which output is generated. The authors posit that there are two-fold benefits from the new skip-GRU cell: coherence between states and addition of a non-linearity. As non-linear functions are extremely useful for complicated mappings, it helps immensely in learning (Liu, Fu, Mei, & Chen, 2017).The authors report that their model scores METEOR scores above their assumed baseline.

2.7 Molding the Attention Models

The attention models described were dependent on the features learned by the network – which means there is still a certain degree of lack of control. On the other hand, there might be a need to modify or mold the attention models to better suit the needs. This leaves the existing attention models wanting. This particular area is being dealt with in this study. The current section describes a few approaches taken in this direction.

There are a few notable studies dealing with correcting or guiding the attention. They provide a good base to start, and employ for this study. Before looking at the aforementioned works, exemplar learning has to be described.

When employing a deep neural network, large dataset is assumed to be available. However, this might not be the case all the time. Exemplar learning helps with this problem by learning the similarity of data with existing knowledge (Bautista, Sanakoyeu, Tikhoncheva & Ommer, 2016). CliqueCNN (Bautista, Sanakoyeu, Tikhoncheva & Ommer, 2016) is an example, where the

authors employ unsupervised exemplar learning for deep neural networks – CNNs in particular. Exemplar learning deals with learning the similarity with existing knowledge, which the authors use for training CNNs with smaller datasets. The unsupervised exemplar learning proposed handles these limitations by updating the similarities and CNNs (Bautista, Sanakoyeu, Tikhoncheva & Ommer, 2016).

Another exemplar approach was employed as a sampling scheme in the paper titled Text-Guided attention model for Image Captioning (Mun, Cho & Han, 2017). Here, the authors propose a new attention model – where the text is used to guide the attention to get better performance. The authors used exemplar approach with captions, and hence the captions used in training are reused for inferencing. The neural network consists of a CNN layer for extracting feature maps, a Skip-Thought Vector (STV) model (Kiros et al., 2015) for pulling the guidance caption, which is then followed by the text-attention model. Finally, an LSTM layer is employed to generate the text. Once the CNN is fed the input image, an STV model fetches candidate captions from the training dataset. The candidate captions are dependent on the visual similarity of the candidate image, as well as the caption consensus scores. Out of these fetched captions, a random caption is sampled using the means described in the previous paper, and used as a guidance caption. The query image is also fed to the CNN layer, which extracts the feature maps, and feeds it to the text-guided attention model along with the guidance caption from the STV. These inputs are weighted (using an attention weight map) and then summed before applying a softmax. Finally, the output of attention is fed as the initial hidden state to the LSTM layer, which generates the caption for the image. The authors report that this network outperforms all the compared models in most, if not all metrics. The authors switch between VGG (Simonyan & Zisserman, 2014), and ResNet (He, Zhang, Ren & Sun, 2016) as the CNN for performance comparisons, of which the ResNet version performs the best.

Yet another approach for ensuring correct attention was proposed by Liu, Mao, Sha and Yuille (2017). Their work, titled Attention Correctness in Neural Image Captioning involves a quantitative metric for measuring the correctness of the attention map. A supervised attention model requires ground truth attention annotations, which are difficult to obtain. Instead, the authors use bounding boxes for each detected object as a general area within which the attention can be

applied. Hence, datasets which contained the bounding box information – Flickr30K (Young at al., 2014) and MSCOCO (Mathews et al., 2016) were used. The authors put forth two kinds of supervised attention models – strong, and weak. The strong supervised attention model requires precise knowledge of the bounding box, and the associated word. Hence, while generating attention maps, the authors (Liu, Mao, Sha & Yuille, 2017) make the weights 0 if the region is not within the expected bounding box. This forces the attention to be within the expected bounding box – thereby giving more accurate results. The weak supervised attention model tackles the case when there isn't a ground truth mapping between bounding boxes, and their associated words. Here, the authors approximate image to language similarity with language to language similarity. The current word, and the class label similarity is used, and the best bounding box is picked. It then follows the same procedure as the strong supervised attention model. For evaluating whether the attention model is within the bounding box, they employ a simple technique. The authors add all the weights within the bounding box, and normalize the value between 0 and 1. Then the attention map which gives the maximum value of the said sum is picked. The authors used the BLEU, and METEOR metrics to evaluate their proposed neural network architecture. The non-supervised attention model (called implicit attention model) is used as baseline, and the authors report that the scores increase consistently after the introduction of the supervised attention model. The authors mention BLEU-4 in particular, citing a 0.9% and 0.7% increase on Flickr30K, and MSCOCO datasets.

Guided attention was another approach worked on by Li, Wu, Peng, Ernst and Fu (2018). The authors created an end-to-end architecture which generated accurate attention maps unlike the usual approaches which result in coarse maps. The aim of the authors for this research was to create as architecture which can generate improved attention maps all the while operating within the constraints of weakly supervised learning. The usual approaches, according to the authors, are not end-to-end, and involve extra work after training the neural network. Then, random parts of the image are hidden while training, thereby forcing the network to learn various different areas of attention for the object. The resulting attention maps are combined later to create the final maps. Another approach the authors mention is the use of two networks. First network used to generate rather coarse attention maps, which are then used to hide those parts in the image. This new composite image is fed to the second network, forcing it to learn to attend on the remaining parts

of the image. Instead of these usual approaches, the authors propose an end-to-end architecture, which consists of two parts, each sharing the same shared weights. For each object class, the first part of the neural network generates the attention map, which are used to get the gradients corresponding to each class. These are used to calculate the inputs to the second part of the network. This input is a mask on the input image, hiding the coarse class. The second network has to now identify the same class using the remnants of the image. The uniqueness however is due to both parts using the same shared weights, and hence trained jointly. Each part contributes to a loss, which are added and then used for the weight updates. This proposed join neural network architecture (Li, Wu, Peng, Ernst & Fu, 2018) outperformed the state of the art, getting accurate segmentation areas.

2.8 Evaluation Metrics

There are 7 metrics which are typically seen in image captioning studies: BLEU 1, BLEU 2, BLEU 3, BLEU 4, ROGUE, METEOR and CIDEr, all described in this section.

BLEU (Bilingual evaluation understudy) was created to measure the quality of text generated by machines when translating text from one language to another. It is a precision metric, reported to correlate highly with human evaluation (Papineni, Roukos, Ward & Zhu, 2002). Hence it has been used to evaluate almost all efforts made in this field. The score is dependent on the n in the n-gram being considered. Hence, usually, BLEU-1, BLEU-2, BLEU-3, and BLEU-4 are reported. The higher n-gram scores correspond to grammatical well-formedness (Banerjee & Lavie, 2005). The scores are generated by comparing the model generated text, and the dataset's captions. However, there has been criticism of the BLEU metric (Callison-Burch, Osborne & Koehn, 2006) – where the authors report that permuting the words according to the bigram gets good scores for sentences with no grammatical sense. Further, Callison-Nurch, Orborne and Koehn (2006) mention that the BLEU metric cannot be guaranteed to correlate with human judgements, and provide evidence where BLEU ranked a poor phrase-based MT system a higher score compared to a good rule-based system (Callison-Burch, Osborne & Koehn, 2006). Hence, few more metrics – ROUGE (Lin, 2004), METEOR (Banerjee & Lavie, 2005) and CIDer (Vedantam, Lawrence & Parikh, 2015) will be used for evaluation as well. All the mentioned metrics are bundled in the MSCOCO evaluation script, and hence can be run simultaneously.

The author of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) package (Lin, 2004) mentions that BLEU was made with evaluation of machine translation in mind, and hence ROUGE was made for better comparison and evaluation of summaries. ROUGE-L, which is used for this task, is a part of the MSCOCO evaluation script. It is used for evaluating how similar sentences are, by considering the Longest Common Subsequence (LCS). It doesn't require consecutive matches, but rather, in-sequence matches. If two sentences are same, the score is 1, and if there is no similarity, the score is 0. One disadvantage put forth by the author (Lin, 2004) is that the metric doesn't consider alternative same length LCSs or shorter LCSs. It should be noted that synonyms are not considered either.

METEOR – Metric for Evaluation of Translation with Explicit Ordering was proposed by Banerjee and Lavie (2005). The authors mention the weakness of the BLEU metric, which depends on n-gram precision. They posit that BLEU doesn't take recall into consideration, which is important when evaluating translations. Apart from the lack of recall, authors mention the use of higher order n-grams, being an indirect measure, when a more direct measure would be better. METEOR metric was made explicitly to address the weaknesses in BLEU, and it can evaluate a translation by computing a score based on word-to-word matches between the translation and the reference (Banerjee & Lavie, 2005).

Finally, CIDer – Consensus based Image Description evaluation (Vedantam, Lawrence & Parikh, 2015) is a relatively newer metric which was made purposefully for evaluating image captions/descriptions. The authors mention that the widely used metrics such as BLEU and ROGUE do not effectively capture human judgement, and have low correlation with it. CIDer uses a consensus mechanism for evaluation, where a number of reference sentences are used to evaluate the candidate sentence. It considers the n-grams which are present in the reference sentences, as well as n-grams which aren't present in the sentences. It also assigns lower weight to n-grams which appear commonly across all the images. Then, TF-IDF (Robertson, 2004) is used to weight the n-grams – both for the occurrence as well as rarity. Finally, cosine similarity of the reference sentences and the candidate sentence is calculated, accounting for both precision and recall. Higher order n-grams are employed to capture richer semantics and grammatical properties.

2.9 Summary

The literature review covers the most relevant scholarly work done in the previous decade which are related to Image captioning. It goes through the datasets which were created, the various advances in image analysis and text generation. Then moving on to combining both and finally application of attention mechanisms and finishing with the evaluation metrics used.

However, none of the work done concentrates on forcing the models to pay attention to a focus object, but rather let the algorithm determine the relevancy. This is what this thesis aims to do – let the researcher specify the focus object on which the algorithm will attend to.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

As mentioned before, current implementations of image captioning neural networks get a free hand at selecting what parts of the image to attend to. This study aims to devise a mechanism to enable researchers to specify which parts of the image to attend and which parts of the said image to ignore. This chapter covers the research framework, the datasets being used, and the evaluation methodology used for this thesis.

3.1 Research framework

For this research study, the datasets and evaluation metrics used in prior work are reused. The training, test, and validation sets were prepared to cater to the updated attention mechanism, as using the datasets as a whole would defeat the point of this study. The baseline was assumed to be the performance of the Show, Attend and Tell (Xu et al., 2015) model, referred to as the implicit soft attention network on the 7 metrics – BLEU 1, BLEU 2, BLEU 3, BLEU 4, METEOR, ROGUE and CIDer.

3.1.1 Network architecture

The network architecture was similar to the Show, Attend and Tell (Xu et al., 2015), where an encoding layer is followed by a decoding layer, with attention in between. This architecture of encoding layer followed by a decoding layer can be considered an archetype of image captioning networks. Google (Sharma, Ding, Goodman & Soricut, 2018) used the Show and Tell (Vinyals et

al., 2015) – which is similar to the Show, Attend and Tell (Xu et al., 2015) (but removes the attention) for benchmarking their new dataset, along with a transformer architecture (Vaswani et al., 2017).

The modification was in the attention mechanism, which forces the attention on the focus object. This modification is explained subsequently.

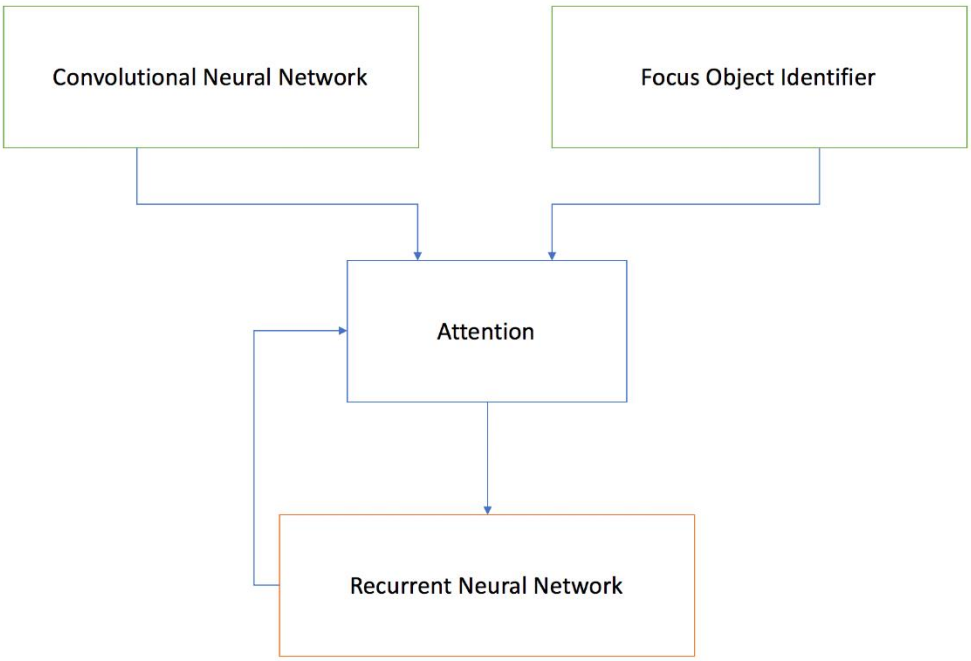

Figure 1: Overall architecture of the neural network.

For this study, the encoding network would be a Convolutional Neural Network (CNN). CNNs have become synonymous with image classification and detection tasks, and they can be modified for use with image captioning. Usually, a CNN consists of a few layers of convolutional layers – each of which have a convolving kernel, followed by a maxpool and a non-linearity. These convolutional layers are then followed by fully connected layers – the plain neural networks. The last layer of these fully connected layers is fed to a soft-max function to get the class prediction.

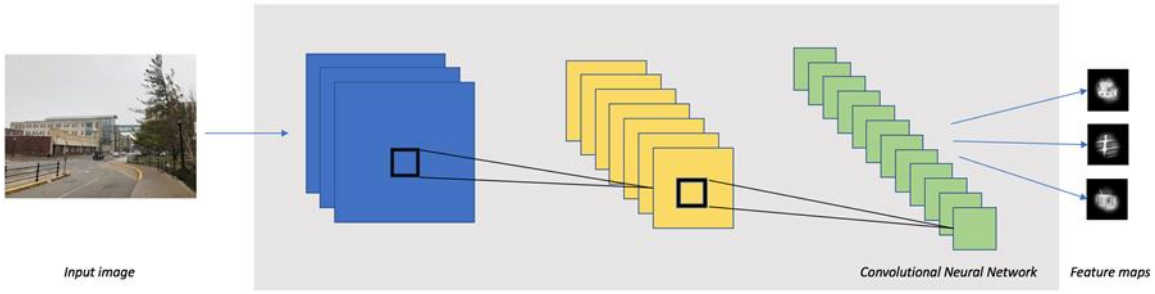

Figure 2: Encoder network - Convolutional Neural Network.

This meant the final output of the network was a fixed length vector – generated by the fully connected layer. These fixed length vectors contained all the information about the detected objects in the image – which may be any number. Image captioning networks without attention typically use this as the input for decoding networks, where the single fixed length vector is used as the context for the text generation. This means that the decoding network has to look at the entirety of the encoding network's output at once, and generate text, resulting in lower emphasis on the individual objects which convey most of the information about the image. However, for this study, the class predictions served no purpose. Hence, the last softmax layer could be trimmed.

There is a second component (referred to as focus object identifier) which is primarily used for identifying the focus object. It is fed to the attention. The output of this network is used for forcing the network to attend to the focus object.

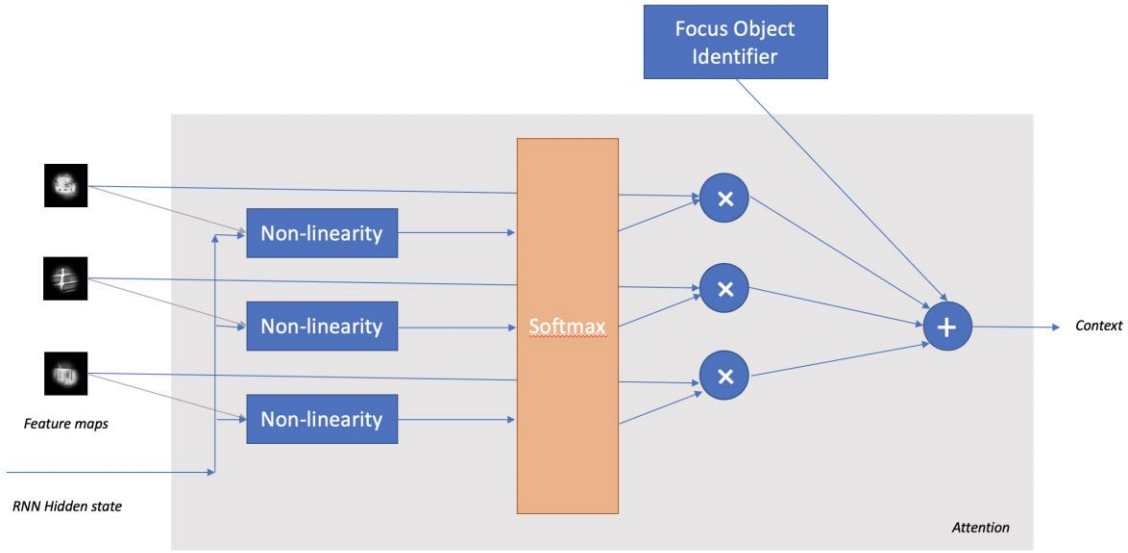

Figure 3: Attention Mechanism

For this study, soft attention was employed, which negated the aforementioned disadvantage. The fully connected layers of the encoder were also trimmed, leaving just the convolutional layers intact. Now, the outputs of the convolutional layers were multiple annotation vectors – each denoting a feature map. With multiple vectors at hand, the decoding network could "look", or "concentrate" on individual vectors, and generate the caption accordingly. Therefore, the output of the convolutional layer was used as the input for the subsequent sections of the network.

Next, these annotation vectors were fed to the attention mechanism. The attention mechanism took the annotation vectors along with a hidden state (part of the decoding layer) to generate the particular region where the network would concentrate. The hidden state vector from the decoder contained information of the previously generated words in the captions, and was constantly updated with each word. Hence, the attention mechanism was constantly generating new regions to concentrate on.

For this study, the attention mechanism was modified – so that the focus object was included in the generated attention regions. To the output of the soft attention, and extracted attention map for a focus object is added. However, this extracted attention map is weighted, so that the output of

the attention does not get saturated. The weighting of the extracted attention map can be static or gradually increased/decreased – giving static and gradual forced attention.

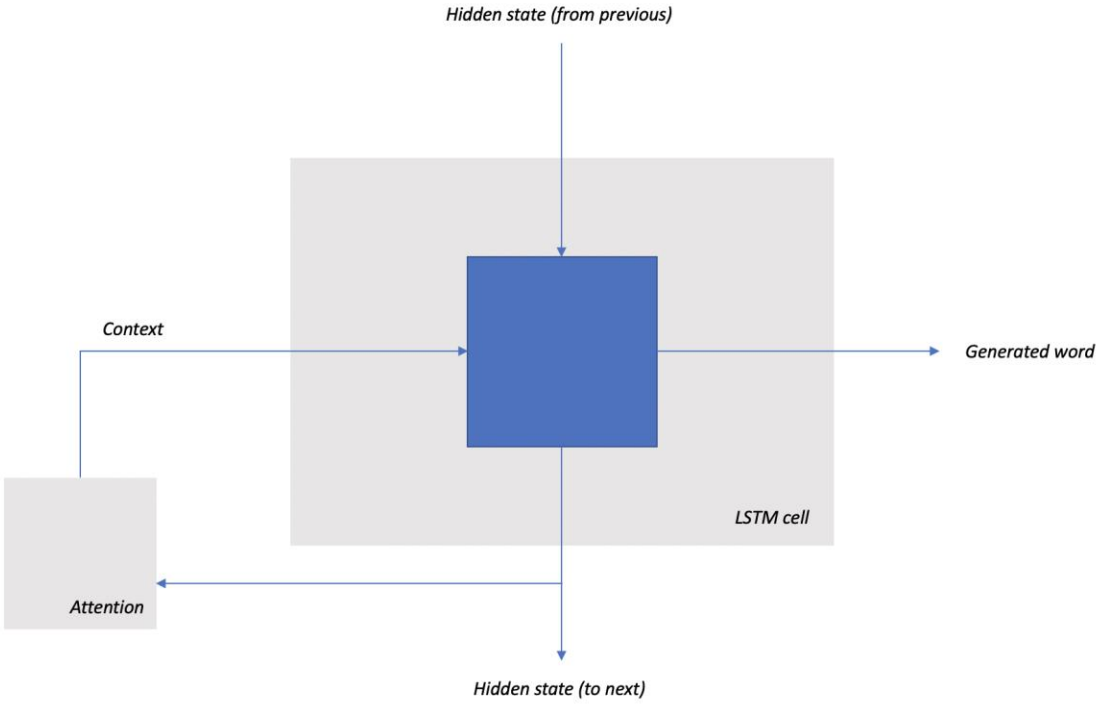

Figure 4: LSTM cells - a set of these comprise the decoder network.

Finally, the decoder network was fed the output of the attention mechanism – which contained the regions to concentrate on. The decoding network was comprised of Recurrent Neural Network (RNN) for this study. The Show, Attend, and Tell architecture uses Long-Short Term Memory (LSTM) cells. The LSTM cells are known for having the ability to "remember" the previously generated words or characters, which is useful for sentence generation. The ability to "remember" is important as sentences are a sequence of words, and hence capturing the essence of the sequence is required. LSTM cells use a hidden state, which is updated with every generated word. The hidden state was also fed to the attention mechanism, so that it could understand how much of the sentence has already been generated. The attention mechanism used the hidden state to creating appropriate attention regions, instead of generating the same region every time. The same configuration of LSTM layers as seen in Show, Attend and Tell (Xu et al., 2015) was used for this study.

Once the network architecture was modelled, and the network was built, it was trained, and then evaluated on the datasets. The metrics chosen for evaluation were employed to measure the performance. For the encoding network, a pre-trained network was used – Resnet50 (He, Zhang, Ren & Sun, 2016). While training, the decoding network was trained, leaving the encoding network as-is.

The Show, Attend, and Tell architecture was chosen as it was an archetype of the encoder-decoder pattern. Studies published recently still use it for benchmarks and baselines (Sharma, Ding, Goodman & Soricut, 2018). Another benefit is the soft attention mechanism, which is simple enough for additions and modifications to be made.

3.1.2 Modification of Attention

The attention mechanism must be able to force the attention on to the selected focus object. This was implemented by the following steps:

1. Extracting the attention map for the corresponding focus object.
2. Using the extracted attention map to force attention while inference.

The first step, extraction of the attention map, involved running the trained baseline model on an image containing a target object. During the inference, the attention maps were sampled, and the map corresponding to the focus object was saved to the disk. This could be done by reading the output of the LSTM network, which generated a word vector. When the generated word vector was referring to the focus object, the corresponding attention map was extracted. Then, the modified architecture, which contains the forced attention was employed for inferencing.

The second step could be implemented in either of the following methods:

- Weighting the extracted attention map, and then adding it to the attention mechanism's output. This is referred to as Static Forced Attention.
    - Here, the extracted attention map cannot be directly added to the attention mechanism's output, as the pixels can saturate (reach maximum value) quickly. This can result in worse performance compared to baseline, as the attention mechanism can no longer attend to specific regions of the image.

- Weighting the extracted attention map, and gradually increasing or decreasing the weights with each time step. This is referred to as Gradual Forced Attention.

Both of these methods were employed and compared during evaluation. The initializing weights for both methods, and the gradual increment or decrement factors were determined empirically. The gradual increment or decrement factor was multiplied to the weights with each time step.

A benefit of this attention modification is that the model does not need to be trained on any of the pruned datasets, no matter what focus object is picked. The baseline model, which was trained on the entirety of the MSCOCO dataset was used. The forcing of the attention happens entirely during inference.

3.2 Dataset

The study used the MSCOCO dataset (Lin et al., 2014) for evaluation. The dataset contained over 328k images with 91 different object categories. Unique objects from top six categories (by number) are used as the focus objects. Each image has five descriptive captions, on which the model was trained. For evaluation, the dataset was split, creating a pruned dataset for each focus object. The network was trained on the entirety of the dataset, but was evaluated on each of the pruned datasets.

For pruning the dataset, the images which contained the focus object label, and its captions which contained the focus object were extracted. This process was done for each of the focus object, thereby resulting in six pruned datasets.

The previous model – Show, Attend and Tell (Xu et al., 2015) used the entire dataset for training and validation. However, for this study, the model was evaluated on the pruned dataset, which was set as the baseline.

3.3 Evaluation

The trained models generated sentences which attempted to describe/caption the image. To evaluate the performance, text-based metrics were used.

As mentioned before, six objects, each from different categories would be used as focus objects, and the network's performance would be measured.

3.4 Testing Methodology

The MSCOCO dataset contains a wide variety of images and their respective captions. It is split into three sets – training, testing and validation. The proposed model however was trained to attend to a specified focus object. Hence, for the evaluation, the MSCOCO dataset had to be pre-processed to remove all instances of images which did not have the selected focus objects.

3.4.1 Pruning the Dataset

The dataset provided the object classes present in each image, along with their captions. Hence the dataset is pruned by searching for images reported to contain the focus object along with the respective captions.

3.4.2 Training the neural network

The baseline neural network architecture used was Show, Attend and Tell (Xu et al., 2015) – which consisted of the classic encoder – decoder network configuration along with attention mechanism. Its encoder network can be either VGG 16 or Resnet 50, of which a pretrained Resnet50 was chosen for this study. Resnet50 (He, Zhang, Ren & Sun, 2016), is a newer convolutional architecture compared to VGG 16 being both better performing as well as less computationally expensive (He, Zhang, Ren & Sun, 2016). The attention mechanism sits between the encoder and decoder, and helps the neural network attend to objects in the image. The outputs of the encoder – the feature maps are extracted and fed to the attention mechanism along with a vector corresponding to the previously generated word by the decoder. Its output, an attention map is then fed to the decoder. The decoder takes the attention map, and a hidden state to generate the word.

While training, only the decoder network with LSTMs is trained, while the pretrained encoder network is left as-is.

The training is done with the following settings:

- Encoder: Resnet50.
- Optimizer: Adam.
- No. of LSTM cells: 512.
- Max. Caption length: 20.
- Attention layers: 2.
- Vocabulary size: 5000
- Beam width: 1
- Epochs: 80.

The neural network was trained on an Intel Xeon Silver 4110 server with dual Nvidia GTX 1080Ti GPUs.

3.4.3 Generation of the caption

This output of the attention mechanism is fed to the decoder network as the context, along with its hidden state. The hidden state contains information about previously generated words, giving the decoder *memory*, helping it to generate better sentences.

3.4.4 Selecting the Focus Objects

The dataset groups objects into categories – animals, vehicles, sports, etc. However, selecting the unique object with the highest occurrence from each of the categories did not create a good evaluation set as some of the unique items listed are a *catch-all*. For example, *sports-ball* is the highest occurring object in the sports category, however, its dataset included tennis balls, footballs, gold balls, etc. Further pruning the dataset yielded much smaller size. Hence, the next largest object was picked – *frisbee*. Upon further classification and pruning, the following objects were picked: dog, train, clock, frisbee, toilet and pizza.

3.4.5 Establishing the baselines

The model is trained on the training set for multiple epochs, and its performance is measured with respect to the validation set after each epoch. The test set is used for validating the performance of the model after going through all epochs.

Once the training is finished, the pruned dataset corresponding to each focus object is used as the evaluation set, and the metrics are calculated. The MSCOCO dataset provides a script for calculating BLEU 1, BLEU 2, BLEU 3, BLEU 4, METEOR, ROUGE-L and CIDer. Hence, all the mentioned metrics are calculated with each evaluation set to establish the baselines.

3.4.6 Extracting attention map for the Focus Object

Once the baselines are established, the next component is the focus object identifier. This provides the means for the attention mechanism to attend to the specified focus object. Here, the attention map for the focus object is extracted from the pre-trained network.

For extracting the attention map for the focus object, each word vector generated is compared to the focus object's word. The attention map corresponding to the focus object's word is extracted when the correct word is generated and saved. As multiple attention maps are saved for each object, they are averaged to get a single attention map for each focus object.

3.4.7 Feeding the attention map to the Attention mechanism

The attention mechanism is modified to take an additional input. Along with the usual inputs (feature maps from encoder, previously generated word vector from decoder), the attention map which was saved is also fed to the mechanism. This attention map helps select the feature map corresponding to the focus object, resulting in the new output which attends to the focus object.

The attention map is added to the output of the attention mechanism. This addition is weighted, allowing the attention map to have less or more effect on the attention mechanism. Having large weights for the attention map can result in malformed outputs as the matrix can get saturated. Here, both the methods of implementing forced attention and employed.

The weights for both static and gradual forced attention were manually selected. For static attention, any weight below 1.0 is selected, and the neural network architecture is evaluated on the dataset. Then, the value is either increased or decreased in steps of 0.05 until a maximum value is reached for all metrics. This is used for static forced attention. For gradual forced attention, the static weight is weighted down at each time step. Any weight above 1.0 will saturate the attention map, which decreases the performance on all metrics.

3.4.8 Evaluating the forced attention neural network

The new neural network with the modified attention mechanism is evaluated on the pruned dataset corresponding to each focus object. Using the MSCOCO evaluation script, the BLEU 1, BLEU 2, BLEU 3, BLEU 4, METEOR, ROUGE-L and CIDer scores are calculated, and compared with the baselines.

The six datasets created for each of the focus objects are further randomly split into five equal parts. Then, both the baseline and the gradual attention architectures are run, and evaluated on the seven metrics. The difference of the evaluation metrics between baseline and proposed neural network architecture are assumed to be normal, and hence Paired T-Test are employed to calculate the statistical significance.

3.5 Summary

This section describes the research framework and methodology used for this study. This includes the hypothesis, the datasets being used, the evaluation metrics used, and the testing methodology. Each of the datasets used are briefly described. The seven metrics employed are also described, after which the testing methodology is expanded upon.

CHAPTER 4. RESULTS AND DISCUSSIONS

The six metrics – BLEU-1, 2, 3, and 4, METEOR, ROUGE and CIDer are calculated for all the focus objects, and compared with the corresponding baselines. Given the encoder and decoder architecture being followed, the performance depended on how and what objects were detected in the images by the encoder. If the focus object was prominent or appeared often, it would be included in the generated caption without needing forced attention. In such cases, depending on the type of forced attention, performance might fall. As the captions generally have more than one object, the objects which have the highest occurrences were easy to focus on. Results for each of the focus object are detailed below, along with their occurrences of the labels in each dataset.

The weights and step value for static and gradual forced attention are determined empirically. Weights are initialized to 0.25 for static forced attention, and varied by 0.1 to get the optimal value (maximum scores in metrics). Then, the optimal value of static forced attention is used for initializing gradual forced attention, along with a step size of 0.5. Here too, both the weights and step size are varied by 0.1 and 0.05 to get the optimal value (maximum scores in metrics).

4.1 Focus Object: Dog

When pruning the dataset for images and captions containing a *dog* (which had the highest unique occurrences among animals), the dataset was left with 4562 results.

4.1.1 Results for Focus Object: Dog with Static Forced Attention

On running the baseline metrics using the unmodified Show, Attend and Tell (Xu et al., 2015) network, and then the forced attention network, all seven metrics saw improvement.

The results for static weight of **0.25** are as follows:

Table 1: Metrics for focus object - dog with static forced attention

Metric	Baseline scores	Static Forced Attn. scores
BLEU - 1	0.582	**0.586**
BLEU – 2	0.409	**0.413**

Table 1 continued

BLEU – 3	0.271	**0.276**
BLEU – 4	0.178	**0.182**
METEOR	0.184	**0.186**
ROGUE	0.445	**0.451**
CIDer	0.491	**0.506**

4.1.2 Results for Focus Object: Dog with Gradual Forced Attention

Using gradual forced attention further improves the scores compared to static forced attention. The results for gradual forced attention starting with **0.45** decreasing with a step size of **0.9** are as follows (statistically significant results are marked with a *):

Table 2: Metrics for focus object – dog with gradual forced attention

Metric	Baseline scores	Grd. Forced Attn. scores
BLEU – 1	0.582	**0.587***
BLEU – 2	0.409	**0.415***
BLEU – 3	0.271	**0.278***
BLEU – 4	0.178	**0.184***
METEOR	0.184	**0.186***
ROGUE	0.445	**0.453***
CIDer	0.491	**0.508***

For the gradual forced attention, Paired T-Test is run to test for statistical significance. The dog dataset is split randomly into 5 equal parts, on which both the baseline and the gradual attention architecture is ran. The results are as recorded below:

Table 3: BLEU - 1 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.581	0.584	0.574	0.585	0.586
Grad. Frc.	0.582	0.587	0.583	0.589	0.594

For the BLEU – 1 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.03**, and the improvements seen are *statistically significant*.

Table 4: BLEU - 2 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.406	0.411	0.399	0.421	0.408
Grad. Frc.	0.411	0.413	0.409	0.422	0.419

For the BLEU – 2 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.046**, and the improvements seen are *statistically significant*.

Table 5: BLEU - 3 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.267	0.269	0.264	0.283	0.273
Grad. Frc.	0.273	0.274	0.272	0.284	0.284

For the BLEU – 3 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.02**, and the improvements seen are *statistically significant*.

Table 6: BLEU - 4 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.177	0.170	0.171	0.189	0.184
Grad. Frc.	0.182	0.176	0.178	0.191	0.192

For the BLEU – 4 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.005**, and the improvements seen are *statistically significant*.

Table 7: METEOR metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.183	0.184	0.184	0.186	0.181
Grad. Frc.	0.187	0.186	0.185	0.188	0.183

For the METEOR metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.01**, and the improvements seen are *statistically significant*.

Table 8: ROGUE L metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.451	0.446	0.441	0.447	0.442
Grad. Frc.	0.457	0.455	0.451	0.449	0.449

For the ROGUE L metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.008**, and the improvements seen are *statistically significant*.

Table 9: CIDer metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.501	0.534	0.505	0.514	0.509
Grad. Frc.	0.533	0.540	0.521	0.536	0.524

For the CIDer metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.013**, and the improvements seen are *statistically significant*.

Statistically Significant Metrics: BLEU – 1, BLEU – 2, BLEU – 3, BLEU -4, METEOR, ROGUE L, and CIDer.
Statistically Insignificant Metrics: None.

4.2 Focus Object: Pizza

When pruning the dataset for images and captions containing a *pizza* (which had the highest unique occurrences among food), the dataset was left with 3319 results.

4.2.1 Results for Focus Object: Pizza with Static Forced Attention

On running the baseline metrics using the unmodified Show, attend and tell (Xu et al., 2015) network, and then the forced attention network, all but one metric saw improvement.

The results for static weight of **0.8** are as follows:

Table 10: Metrics for focus object – pizza with static forced attention

Metric	Baseline scores	Static Forced Attn. scores
BLEU - 1	0.632	**0.633**
BLEU – 2	0.458	**0.460**
BLEU – 3	0.314	**0.316**
BLEU – 4	0.213	**0.214**
METEOR	0.198	0.198
ROGUE	0.469	**0.473**
CIDer	0.362	**0.366**

4.2.2 Results for Focus Object: Pizza with Gradual Forced Attention

Using gradual forced attention further improves the scores compared to static forced attention. Here, all 7 metrics see improvement compared to baseline.

The results for gradual forced attention starting with **0.75** decreasing with a step size of **0.25** are as follows (statistically significant results are marked with a *):

Table 11: Metrics for focus object – pizza with gradual forced attention

Metric	Baseline scores	Grd. Forced Attn. scores
BLEU - 1	0.632	**0.633**
BLEU – 2	0.458	**0.462***
BLEU – 3	0.314	**0.319***
BLEU – 4	0.213	**0.219***
METEOR	0.198	**0.200**
ROGUE	0.469	**0.475***
CIDer	0.362	**0.368***

For the gradual forced attention, Paired T-Test is run to test for statistical significance. The pizza dataset is now split randomly into 5 equal parts, on which both the baseline and the gradual attention architecture is ran. The results are as recorded below:

Table 12: BLEU - 1 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.655	0.626	0.614	0.636	0.629
Grad. Frc.	0.661	0.627	0.612	0.637	0.638

For the BLEU – 1 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.2**, and the improvements seen are ***not*** *statistically significant*.

Table 13: BLEU - 2 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.478	0.449	0.440	0.468	0.457
Grad. Frc.	0.494	0.461	0.451	0.481	0.471

For the BLEU – 2 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.0001**, and the improvements seen are *statistically significant*.

Table 14: BLEU - 3 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.327	0.303	0.299	0.327	0.313
Grad. Frc.	0.349	0.320	0.318	0.346	0.330

For the BLEU – 3 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.00003**, and the improvements seen are *statistically significant*.

Table 15: BLEU - 4 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.221	0.205	0.202	0.229	0.209
Grad. Frc.	0.238	0.218	0.221	0.245	0.222

For the BLEU – 4 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.0001**, and the improvements seen are *statistically significant*.

Table 16: METEOR metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.205	0.194	0.192	0.203	0.195
Grad. Frc.	0.209	0.193	0.189	0.205	0.199

For the METEOR metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.4**, and the improvements seen are ***not*** *statistically significant*.

Table 17: ROGUE L metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.477	0.465	0.461	0.478	0.466
Grad. Frc.	0.489	0.477	0.468	0.495	0.482

For the ROGUE L metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.0019**, and the improvements seen are *statistically significant*.

Table 18: CIDer metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.379	0.349	0.338	0.457	0.400
Grad. Frc.	0.432	0.373	0.381	0.522	0.423

For the CIDer metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.007**, and the improvements seen are *statistically significant*.

Statistically Significant Metrics: BLEU – 2, BLEU – 3, BLEU – 4, ROGUE L, and CIDer.
Statistically Insignificant Metrics: BLEU – 1, and METEOR.

4.3 Focus Object: Frisbee

When pruning the dataset for images and captions containing a *frisbee,* the dataset was left with 2268 results. The frisbee images usually had a person or multiple people in the picture.

4.3.1 Results for Focus Object: Frisbee with Static Forced Attention

On running the baseline metrics using the unmodified Show, attend and tell (Xu et al., 2015) network, and then the forced attention network, 4 metrics saw improvement, while the remaining remained the same.

The results for static weight of **0.75** are as follows:

Table 19: Metrics for focus object – frisbee with static forced attention

Metric	Baseline scores	Static Forced Attn. scores
BLEU - 1	0.564	0.564
BLEU – 2	0.385	**0.387**
BLEU – 3	0.256	**0.260**
BLEU – 4	0.174	**0.176**
METEOR	0.204	0.204
ROGUE	0.451	**0.454**
CIDer	0.463	**0.473**

4.3.2 Results for Focus Object: Frisbee with Gradual Forced Attention

Using gradual forced attention further improves the scores compared to static forced attention. Again, all 7 metrics see improvement compared to baseline.

The results for gradual forced attention starting with **0.8** decreasing with a step size of **0.33** are as follows: (statistically significant results are marked with a *)

Table 20: Metrics for focus object – frisbee with gradual forced attention

Metric	Baseline scores	Grd. Forced Attn. scores
BLEU - 1	0.564	**0.566**
BLEU – 2	0.385	**0.388**
BLEU – 3	0.256	**0.262**
BLEU – 4	0.174	**0.179***
METEOR	0.204	**0.206**
ROGUE	0.451	**0.457***

Table 20 continued

CIDer	0.463	**0.484***

For the gradual forced attention, Paired T-Test is run to test for statistical significance. The frisbee dataset is now split randomly into 5 equal parts, on which both the baseline and the gradual attention architecture is ran. The results are as recorded below:

Table 21: BLEU - 1 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.553	0.573	0.580	0.563	0.550
Grad. Frc.	0.556	0.573	0.593	0.559	0.548

For the BLEU – 1 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.539**, and the improvements seen are ***not*** *statistically significant*.

Table 22: BLEU - 2 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.375	0.384	0.409	0.384	0.375
Grad. Frc.	0.376	0.386	0.425	0.381	0.373

For the BLEU – 2 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.459**, and the improvements seen are ***not*** *statistically significant*.

Table 23: BLEU - 3 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.247	0.245	0.288	0.250	0.251
Grad. Frc.	0.254	0.253	0.301	0.252	0.251

For the BLEU – 3 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.059**, and the improvements seen are ***not*** *statistically significant*.

Table 24: BLEU - 4 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.168	0.164	0.202	0.167	0.170
Grad. Frc.	0.174	0.172	0.210	0.171	0.170

For the BLEU – 4 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.025**, and the improvements seen are *statistically significant*.

Table 25: METEOR metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.202	0.208	0.207	0.207	0.197
Grad. Frc.	0.203	0.209	0.212	0.205	0.200

For the METEOR metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.241**, and the improvements seen are ***not*** *statistically significant*.

Table 26: ROGUE L metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.450	0.453	0.456	0.457	0.438
Grad. Frc.	0.456	0.459	0.468	0.459	0.441

For the ROGUE L metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.029**, and the improvements seen are *statistically significant*.

Table 27: CIDer metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.499	0.505	0.501	0.495	0.521
Grad. Frc.	0.507	0.534	0.548	0.517	0.532

For the CIDer metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.028**, and the improvements seen are *statistically significant*.

Statistically Significant Metrics: BLEU – 4, ROGUE L, and CIDer.

Statistically Insignificant Metrics: BLEU – 1, BLEU – 2, BLEU – 3, and METEOR.

4.4 Focus Object: Clock

When pruning the dataset for images and captions containing a *clock*, the dataset was left with 4863 results.

4.4.1 Results for Focus Object: Clock with Static Forced Attention

On running the baseline metrics using the unmodified Show, attend and tell (Xu et al., 2015) network, and then the forced attention network, 3 metrics saw improvement, 3 had decrement, and one metric had the no change in performance.

The results for static weight of **0.85** are as follows:

Table 28: Metrics for focus object – clock with static forced attention

Metric	Baseline scores	Static Forced Attn. scores
BLEU - 1	**0.608**	0.605
BLEU – 2	0.463	0.463
BLEU – 3	0.351	**0.352**
BLEU – 4	0.254	**0.256**
METEOR	**0.195**	0.193
ROGUE	**0.485**	0.484
CIDer	0.370	**0.372**

4.4.2 Results for Focus Object: Clock with Gradual Forced Attention

Using gradual forced attention further improves the scores compared to static forced attention. Again, all 7 metrics see improvement compared to baseline.

The results for gradual forced attention starting with **1** decreasing with a step size of **0.2** are as follows (statistically significant results are marked with a *):

Table 29: Metrics for focus object – clock with gradual forced attention

Metric	Baseline scores	Grd. Forced Attn. scores
BLEU - 1	0.608	**0.611**
BLEU – 2	0.463	**0.469**
BLEU – 3	0.351	**0.356***
BLEU – 4	0.254	**0.259**
METEOR	0.195	**0.198***
ROGUE	0.485	**0.489**
CIDer	0.370	**0.380**

For the gradual forced attention, Paired T-Test is run to test for statistical significance. The clock dataset is now split randomly into 5 equal parts, on which both the baseline and the gradual attention architecture is ran. The results are as recorded below:

Table 30: BLEU - 1 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.600	0.600	0.626	0.607	0.606
Grad. Frc.	0.600	0.606	0.623	0.609	0.616

For the BLEU – 1 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.25**, and the improvements seen are ***not*** *statistically significant*.

Table 31: BLEU - 2 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.452	0.461	0.478	0.469	0.456
Grad. Frc.	0.456	0.468	0.478	0.474	0.469

For the BLEU – 2 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.052**, and the improvements seen are ***not*** *statistically significant*.

Table 32: BLEU - 3 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.341	0.354	0.360	0.357	0.342
Grad. Frc.	0.344	0.362	0.361	0.361	0.354

For the BLEU – 3 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.046**, and the improvements seen are *statistically significant*.

Table 33: BLEU - 4 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.248	0.259	0.259	0.259	0.244
Grad. Frc.	0.249	0.266	0.259	0.264	0.257

For the BLEU – 4 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.089**, and the improvements seen are ***not*** *statistically significant*.

Table 34: METEOR metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.195	0.193	0.196	0.196	0.193
Grad. Frc.	0.199	0.195	0.197	0.198	0.199

For the METEOR metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.028**, and the improvements seen are *statistically significant*.

Table 35: ROGUE L metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.484	0.483	0.488	0.484	0.488
Grad. Frc.	0.484	0.490	0.490	0.487	0.496

For the ROGUE L metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.057**, and the improvements seen are ***not*** *statistically significant*.

Table 36: CIDer metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.427	0.419	0.395	0.365	0.361
Grad. Frc.	0.431	0.436	0.399	0.355	0.395

For the CIDer metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.25**, and the improvements seen are ***not*** *statistically significant*.

BLEU – 1, Meteor and ROGUE see decrease in scores, while BLEU – 3, 4, and CIDer see improvement on this dataset when static forced attention is used. Using gradual forced attention improves these results, so there is no decrease in the seven metrics.

Statistically Significant Metrics: BLEU – 3, METEOR.
Statistically Insignificant Metrics: BLEU – 1, BLEU – 2, BLEU – 4, ROGUE L, and CIDer.

4.4.3 Discussion for Focus Object: Clock

This is the last object which shows improvements in all metrics when using gradual forced attention, and the last object in which more than one metrics show statistically significant improvements. However, the margin of improvements of five metrics fall short of being statistically significant. To understand this, a few images from the clock dataset are randomly sampled, and their captions are extracted.

Figure 5: Randomly sampled image from clock dataset (MSCOCO (Lin at al., 2014))

Captions in the training dataset for the sampled figure:

- a kitchen with white cupboards, a bowl of bananas, cookbooks, a blue teakettle and other sundries.
- a kitchen filled with lots of pots, pans and dishes.
- shelves in the kitchen filled with books, cups, glasses and a clock
- an area of a kitchen with the stove, oven and shelves with books
- a kitchen counter top with many different appliances.

While training the forced attention model, the entire dataset is used, and clock appears in fewer captions, and hence the overall score is reduced due to fewer matching n-grams.

Similar occurrence on yet another randomly sampled image:

Figure 6: Randomly sampled image from clock dataset (MSCOCO (Lin at al., 2014))

The reference captions for this image are:

- woman sitting beside table posing for picture with a smile.
- woman in big hoop dress sitting down at a chair with a clock.
- a woman is sitting in her chair posing.
- a very old picture of a women posing.
- an old photo showing a woman near a clock.

Here too, the number of matching n-grams is reduced due to more captions not having the focus object.

4.5 Focus Object: Train

When pruning the dataset for images and captions containing a *train*, the dataset was left with 3745 results.

4.5.1 Results for Focus Object: Train with Static Forced Attention

On running the baseline metrics using the unmodified Show, attend and tell (Xu et al., 2015) network, and then the forced attention network, 4 metrics saw a decrease in performance. Two stayed the same, while one increased.

The results for static weight of **0.75** are as follows:

Table 37: Metrics for focus object – train with Static Forced Attention

Metric	Baseline scores	Static Forced Attn. scores
BLEU - 1	**0.587**	0.585
BLEU – 2	**0.392**	0.391
BLEU – 3	**0.235**	0.234
BLEU – 4	0.135	0.135
METEOR	**0.199**	0.198
ROGUE	0.445	0.445
CIDer	0.301	**0.303**

4.5.2 Results for Focus Object: Train with Gradual Forced Attention

Using gradual forced attention further improves the scores compared to static forced attention. The decrement in scores seen with static forced attention is eliminated. Further, 4 metrics see improvement compared to baselines.

The results for gradual forced attention starting with **0.7** decreasing with a step size of **0.25** are as follows (statistically significant results are marked with a *):

Table 38: Metrics for focus object – train with gradual forced attention

Metric	Baseline scores	Grd. Forced Attn. scores
BLEU - 1	0.587	0.587
BLEU – 2	0.392	**0.393**
BLEU – 3	0.235	**0.237**
BLEU – 4	0.135	**0.138***
METEOR	0.199	0.199
ROGUE	0.445	0.445
CIDer	0.301	**0.304**

For the gradual forced attention, Paired T-Test is run to test for statistical significance. The train dataset is now split randomly into 5 equal parts, on which both the baseline and the gradual attention architecture is ran. The results are as recorded below:

Table 39: BLEU - 1 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.582	0.592	0.604	0.595	0.562
Grad. Frc.	0.578	0.589	0.606	0.596	0.564

For the BLEU – 1 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.771**, and the improvements seen are ***not*** *statistically significant*.

Table 40: BLEU - 2 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.396	0.396	0.404	0.397	0.367
Grad. Frc.	0.395	0.395	0.410	0.398	0.369

For the BLEU – 2 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.338**, and the improvements seen are ***not*** *statistically significant*.

Table 41: BLEU - 3 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.246	0.238	0.235	0.238	0.215
Grad. Frc.	0.248	0.236	0.242	0.239	0.217

For the BLEU – 3 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.239**, and the improvements seen are ***not*** *statistically significant*.

Table 42: BLEU - 4 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.147	0.139	0.133	0.132	0.124
Grad. Frc.	0.150	0.142	0.136	0.135	0.125

For the BLEU – 4 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.002**, and the improvements seen are *statistically significant*.

Table 43: METEOR metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.201	0.200	0.205	0.203	0.188
Grad. Frc.	0.200	0.196	0.207	0.203	0.188

For the METEOR metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.573**, and the improvements seen are ***not*** *statistically significant*.

Table 44: ROGUE L metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.451	0.439	0.451	0.448	0.435
Grad. Frc.	0.450	0.438	0.454	0.449	0.434

For the ROGUE L metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.814**, and the improvements seen are ***not*** *statistically significant*.

Table 45: CIDer metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.346	0.309	0.294	0.300	0.329
Grad. Frc.	0.347	0.210	0.299	0.318	0.324

For the CIDer metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.357**, and the improvements seen are ***not*** *statistically significant*.

When using static forced attention, BLEU – 1, 2, 3, and Meteor see decrease in scores, while BLEU-4 and ROGUE stay the same. Only CIDer sees improvement on this dataset. On switching to gradual forced attention, there is no decrement in performance, and BLEU – 2, 3, 4, and CIDer metrics see improvements.

Statistically Significant Metrics: BLEU – 4.
Statistically Insignificant Metrics: BLEU – 1, BLEU – 2, BLEU – 3, METEOR, ROGUE L, and CIDer.

4.5.3 Discussion for Focus Object: Train

By randomly sampling images from the train dataset, it is possible to understand the poor performance of the forced attention architecture.

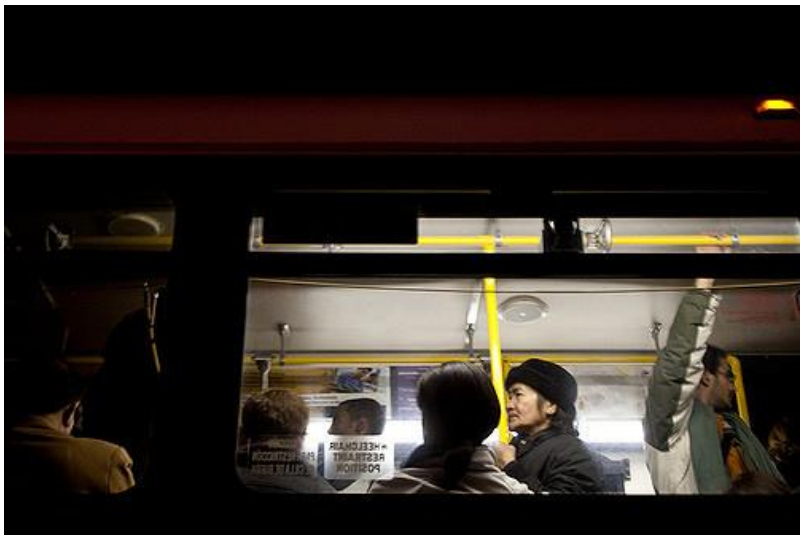

Figure 7: Randomly sampled image from train dataset (MSCOCO (Lin at al., 2014))

Generated caption: *A group of people sitting on a table.*

The above image is included in the train dataset as it is a picture from within the train. However, none of the typical features when looking from outside are seen, and hence, the architecture isn't able to mention *train* even when forced. As forcing attention involves modifying the attention maps themselves, decrement in performance is seen when static forced attention is employed.

Another kind of issue can be explained with this image

Figure 8: Randomly sampled image from train dataset (MSCOCO (Lin at al., 2014))

The dataset's caption for this image included: *A steam engine that is travelling down some track*.

Here, a steam engine can mean the train object itself. However, information is not provided to the neural network during the training procedure. Hence, this can prove to be another source of decrement of performance.

4.6 Focus Object: Toilet

When pruning the dataset for images and captions containing a *toilet*, the dataset was left with 3502 results.

4.6.1 Results for Focus Object: Toilet with Static Forced Attention

On running the baseline metrics using the unmodified Show, attend and tell (Xu et al., 2015) network, and then the forced attention network, all metrics saw a decrease in performance.

The results for static weight of **0.25** are as follows:

Table 46: Metrics for focus object – toilet with static forced attention

Metric	Baseline scores	Static Forced Attn. scores
BLEU - 1	**0.670**	0.668
BLEU – 2	**0.540**	0.537
BLEU – 3	**0.412**	0.408
BLEU – 4	**0.313**	0.309
METEOR	**0.228**	0.227
ROGUE	**0.552**	0.551
CIDer	**0.453**	0.442

4.6.2 Results for Focus Object: Toilet with Gradual Forced Attention

Using gradual forced attention further improves the scores compared to static forced attention. The decrement in scores seen with static forced attention is eliminated. Further, 2 metrics see improvement compared to baselines.

The results for gradual forced attention starting with **0.15** decreasing with a step size of **0.15** are as follows (statistically significant results are marked with a *):

Table 47: Metrics for focus object – toilet with gradual forced attention

Metric	Baseline scores	Grd. Forced Attn. scores
BLEU - 1	0.670	0.670
BLEU – 2	0.540	0.540
BLEU – 3	0.412	0.412
BLEU – 4	0.313	**0.314**
METEOR	0.228	0.228
ROGUE	0.552	0.552
CIDer	0.453	**0.454**

For the gradual forced attention, Paired T-Test is run to test for statistical significance. The toilet dataset is now split randomly into 5 equal parts, on which both the baseline and the gradual attention architecture is ran. The results are as recorded below:

Table 48: BLEU - 1 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.664	0.651	0.689	0.677	0.668
Grad. Frc.	0.664	0.651	0.688	0.678	0.669

For the BLEU – 1 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.621**, and the improvements seen are ***not*** *statistically significant*.

Table 49: BLEU - 2 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.531	0.509	0.566	0.552	0.540
Grad. Frc.	0.530	0.509	0.565	0.553	0.542

For the BLEU – 2 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.748**, and the improvements seen are ***not*** *statistically significant*.

Table 50: BLEU - 3 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.404	0.372	0.441	0.428	0.413
Grad. Frc.	0.403	0.372	0.440	0.429	0.413

For the BLEU – 3 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.621**, and the improvements seen are ***not*** *statistically significant*.

Table 51: BLEU - 4 metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.306	0.267	0.346	0.329	0.316
Grad. Frc.	0.306	0.267	0.346	0.330	0.317

For the BLEU – 4 metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.177**, and the improvements seen are ***not*** *statistically significant*.

Table 52: METEOR metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.224	0.218	0.239	0.235	0.226
Grad. Frc.	0.224	0.218	0.239	0.236	0.227

For the METEOR metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.177**, and the improvements seen are ***not*** *statistically significant*.

Table 53: ROGUE L metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.546	0.530	0.571	0.563	0.549
Grad. Frc.	0.545	0.530	0.571	0.564	0.550

For the ROGUE L metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.621**, and the improvements seen are ***not*** *statistically significant*.

Table 54: CIDer metric for baseline and gradual forced attention arch.

	Split 1	Split 2	Split 3	Split 4	Split 5
Baseline	0.465	0.486	0.513	0.474	0.463
Grad. Frc.	0.465	0.486	0.512	0.479	0.464

For the CIDer metric, on running the Paired T test with significance set to 0.05, the p value is calculated to be **0.394**, and the improvements seen are ***not*** *statistically significant*.

As seen above, all metrics saw a decrease in performance when using static forced attention. On switching to gradual forced attention, there is no decrement in performance, and the BLEU – 4 and CIDer scores show minor improvements.

Statistically Significant Metrics: None.

Statistically Insignificant Metrics: BLEU – 1, BLEU – 2, BLEU – 3, BLEU – 4, METEOR, ROGUE L, and CIDer.

4.6.3 Discussion for Focus Object: Toilet

To understand the lack of significant improvements for this particular focus object, the few images from this dataset are randomly sampled.

Figure 9: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014))

Figure 10: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014))

Figure 11: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014))

Figures 9 and 10 show the first issue with the dataset – visually and textually distinct objects within the toilet dataset. Microsoft categorized this into the toilets section, though the captions use *urinal* to refer to the toilets. As the neural network was trained on the entire dataset, the attention map used for forcing toilet did not correspond to the objects seen here.

Another issue is seen in Figure 11, where the toilet is barely visible, and is does not contain visually distinct features seen in a toilet, whose typical example is seen in Figure 10.

Figure 12: Randomly sampled image from toilet dataset (MSCOCO (Lin at al., 2014))

Apart from images, this dataset contains *urinals, restrooms, sink* and *bathrooms* as reference to toilet instead of the word *toilet* itself. This too presents another problem for the forced attention architecture.

4.7 Significance of the improvements

Though the improvements and regressions in the metrics seem insignificant given the low margin, this is seen in quite a few studies published in the space. Text-Guided attention model for Image Captioning (Mun, Cho & Han, 2017) in AAAI - 2017 reports improvements of similar margin. Their new approach results in 0.012, 0.010, 0.007, 0.005, 0.001, 0.044 increase in BLEU – 1, 2, 3, 4, METEOR and CIDer metrics. Similarly, knowing when to look (Lu, Xiong, Parikh & Socher, 2016) in CVPR - 2017 reports similar improvements of 0.009, 0.009, 0.008, 0.006, 0.008, 0.008, 0.058 in BLEU – 1, 2, 3, 4, METEOR, ROGUE-L and CIDer metrics.

CHAPTER 5. CONCLUSION AND FUTURE WORK

Adding the focus object's attention map to the attention mechanism's output is able to force the attention of the neural network. However, the results seen for the selected focus objects are not consistent. For the first two objects – dog and pizza, more than half the metrics show statistically significant improvements. For the frisbee, and clock, the number of metrics which show statistically significant improvements fall. For the train, only one metric has statistically significant improvement, while toilet sees no statistically significant improvements.

Four of the focus objects (dog, pizza, frisbee and the clock) show improvements in all seven evaluation metrics, although not all are statistically significant. The remaining two objects – train and toilet show improvements in fewer metrics. The train shows improvements in four out of the seven metrics, while the toilet shows improvements in two out of the seven metrics.

The proposed neural network architecture does not perform consistently over the six selected focus objects. Hence, focus objects on which the proposed neural network architecture performs poorly are further investigated. For three of the focus objects, few images and their captions are randomly sampled and examined. Various potential causes are looked at.

The issues identified with the approach taken in this study can be summarized as follows:

- Unequal number of reference captions which contain, and do not contain the focus object.
- Visually and textually distinct focus objects in the datasets.
- Use of synonyms in the datasets while referring to the focus object.
- Visually ambiguous focus objects in the dataset.

5.1 Future Work

Fixing each of the identified issues can improve the results further and might get statistically significant results.

To fix the first problem, the dataset pruning has to be improved. While preparing the dataset, the number of captions which contain the focus object, and those which do not contain the focus object have to be normalized. This may result in discarding captions, but it will help balance the dataset.

The second issue is the presence of visually and textually distinct focus objects. In this case, the visually distinct objects can be handled by keeping track of multiple attention maps for the selected focus objects. The procedure to do this can be automated as well, by measuring the visual similarity between each attention map.

The textually distinct focus objects and synonyms are a related issue, where a single focus object is referred to by different words in the dataset. This can be fixed during the preparation of the dataset, where the synonyms are swapped with the focus object's name.

Finally, to handle visually ambiguous focus objects, newer image encoding networks can be used. For this study, the Resnet50 (He, Zhang, Ren & Sun, 2016) is employed. Newer networks such as capsule networks are view-point invariant (Sabour, Frosst, and Hinton, 2017) might be able to tackle this particular issue and improve the performance.

5.2 Final Words

The proposed forced attention model has a few benefits, such as

- Not needing retraining of the Image encoder or Caption Generator for each focus object.
- Able to be completely automated for a focus object.
- Use pre-trained publicly available encoder networks.

However, there is much scope for improvement. This is seen with the lack of improvement of the seven metrics for all focus objects. Various potential causes are explained, and means to address them are mentioned. The neural network architectures seen in this space can be reused in other applications. Hence, improvements to this architecture to make it perform better on any focus object would help in not just image captioning, but in video analysis, image tagging, extracting context of objects, etc.

REFERENCES

Aditya, S. (2017). Explainable Image Understanding Using Vision and Reasoning. In *AAAI* (pp. 5028-5029).

Aditya, S., Baral, C., Yang, Y., Aloimonos, Y. & Fermuller, C. (2016). DeepIU: An Architecture for Image Understanding. In *Advances of Cognitive Systems*.

Aditya, S., Yang, Y., Baral, C., Fermuller, C., & Aloimonos, Y. (2015, March). Visual common-sense for scene understanding using perception, semantic parsing and reasoning. In *2015 AAAI Spring Symposium Series*.

Aker, A. & Gaizauskas, R. (2009). Summary Generation for Toponym-Referenced Images using Object Type Language Models. *International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Aker, A., & Gaizauskas, R. (2010, July). Generating image descriptions using dependency relational patterns. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1250-1258). Association for Computational Linguistics.

Amazon Mechanical Turk. (2018). Mturk.com. Retrieved 18 November 2018, from https://www.mturk.com/.

Ba, J., Mnih, V., & Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Banerjee, S., & Lavie, A. (2005). 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* (pp. 65-72).

Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1), 3-31.

Bautista, M. A., Sanakoyeu, A., Tikhoncheva, E., & Ommer, B. (2016). Cliquecnn: Deep unsupervised exemplar learning. In *Advances in Neural Information Processing Systems* (pp. 3846-3854).

Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*.

Callison-Burch, C., Osborne, M., & Koehn, P. (2006, April). Re-evaluation the Role of Bleu in Machine Translation Research. In EACL (Vol. 6, pp. 249-256).

Chen, M., Ding, G., Zhao, S., Chen, H., Liu, Q., & Han, J. (2017, February). Reference Based LSTM for Image Captioning. In *AAAI* (pp. 3981-3987).

Chen, X., & Lawrence Zitnick, C. (2015). Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2422-2431).

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655).

Duygulu, P., Barnard, K., de Freitas, J. F., & Forsyth, D. A. (2002, May). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision* (pp. 97-112). Springer, Berlin, Heidelberg.

Elhoseiny, M., Cohen, S., Chang, W., Price, B. L., & Elgammal, A. M. (2017, February). Sherlock: Scalable Fact Learning in Images. In *AAAI* (pp. 4016-4024).

Fang, H., Gupta, S., Iandola, F., Srivastava, R. K., Deng, L., Dollár, P., ... & Lawrence Zitnick, C. (2015). From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1473-1482).

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47, 853-899.

Jones, N. (2014). Computer science: The learning machines. *Nature News*, 505(7482), 146.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3128-3137).

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294-3302).

Klein, D., & Manning, C. D. (2003, July). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 423-430). Association for Computational Linguistics.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., ... & Berg, T. L. (2013). Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12), 2891-2903..

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Li, K., Wu, Z., Peng, K. C., Ernst, J., & Fu, Y. (2018). Tell me where to look: Guided attention inference network. *arXiv preprint arXiv:1802.10171*.

Li, L., Tang, S., Deng, L., Zhang, Y., & Tian, Q. (2017). Image Caption with Global-Local Attention. In *AAAI* (pp. 4133-4139).

Lin, C. Y., (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Proc. of the Workshop on Text Summarization Branches Out* (pp. 25–26).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

Liu, C., Mao, J., Sha, F., & Yuille, A. L. (2017). Attention Correctness in Neural Image Captioning. In *AAAI* (pp. 4176-4182).

Liu, Y., Fu, J., Mei, T., & Chen, C. W. (2017, February). Let Your Photos Talk: Generating Narrative Paragraph for Photo Stream via Bidirectional Attention Recurrent Neural Networks. In *AAAI* (pp. 1445-1452).

Lu, J., Xiong, C., Parikh, D., & Socher, R. (2016). Knowing when to look: Adaptive attention via A visual sentinel for image captioning. *arXiv preprint arXiv:1612.01887*.

Ma, L., Lu, Z., & Li, H. (2016, February). Learning to Answer Questions from Image Using Convolutional Neural Network. In *AAAI* (Vol. 3, No. 7, p. 16).

Maind, S. B., & Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, *2*(1), 96-100.

Malinowski, M., & Fritz, M. (2014). A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems* (pp. 1682-1690)..

Mathews, A. P., Xie, L., & He, X. (2016, February). SentiCap: Generating Image Descriptions with Sentiments. In AAAI (pp. 3574-3580).

Mun, J., Cho, M., & Han, B. (2017). Text-Guided Attention Model for Image Captioning. In *AAAI* (pp. 4233-4239).

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011, December). Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning* (Vol. 2011, No. 2, p. 5).

Ordonez, V., Kulkarni, G., & Berg, T. L. (2011). Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems* (pp. 1143-1151).

Pan, J. Y., Yang, H. J., Faloutsos, C., & Duygulu, P. (2004, June). Gcap: Graph-based automatic image captioning. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (pp. 146-146). IEEE.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311-318). Association for Computational Linguistics.

Ren, M., Kiros, R., & Zemel, R. (2015). Exploring models and data for image question answering. In *Advances in neural information processing systems* (pp. 2953-2961).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation*, *60*(5), 503-520.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211-252.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866).

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Sharma, P., Ding, N., Goodman, S., & Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 2556-2565).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 1017-1024)

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE, 105*(12), 2295-2329.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*(pp. 5998-6008).

Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4566-4575).

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).

Ward, L. (2008). Attention. Scholarpedia, 3(10), 1538. doi:10.4249/scholarpedia.1538.

Wu, Z., & Palmer, M. (1994, June). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 133-138). Association for Computational Linguistics.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. S., & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning* (pp. 2048-2057).

Young, P., Lai, A., Hodosh, M., & Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2, 67-78.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.