A MINI PROJECT REPORT ON

Image Captioning Using Deep Neural Networks

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Computer Engineering

by

Mrunal S Jadhav(16102030)
Aditya G Joshi(16102017)

Under the Guidance of

Prof. Sachin Malave

Department of Computer Engineering
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled ***"Image Captioning Using Deep Neural Networks"*** Submitted by ***"Mrunal S Jadhav"(16102030), "Aditya G Joshi"(16102017),***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Computer Engineering*** from ***University of Mumbai***.

Prof. Sachin Malave
Guide
Head Department of Computer Engineering

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Image Captioning Using Deep Neural Networks"*** submitted by ***"Mrunal S Jadhav" (16102030),"Aditya G Joshi" (16102017)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Engineering***,to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020.

Prof. Sachin Malave	Dr. Uttam D.Kolekar
Guide	Principal
Head Department of Computer Engineering	

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Mrunal S Jadhav 16102030)
(Aditya G Joshi 16102017)

Date:

Contents

List of Figures

List of Tables

Abstract

The visual world is populated with a vast number of objects, the most appropriate labeling of which is often ambiguous, task specific, or admits multiple equally correct answers. A quick glance is sufficient for a human to understand and describe what is happening in the picture. The quest of connecting computer vision and natural language processing is a long way of touching the holy grail in artificial intelligence.

Image captioning is the task of generating natural language utterances based on the understanding of visual snippets of a scene.Image captioning is a much more involved task than image recognition or classification, because of the additional challenge of learning representations of the interdependence between the objects/concepts in the image and the creation of a succinct sentential narration.

The task is to transform a sentence S written in its source language,into its translation T in the target language, by maximising the probability P(T | S).Aided by the advances in training neural networks and large dataset available, there has been a surge in research interest attacking the image caption generation problem. A combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) which seeks to progress directly from image features to text can be progressed to define a single end-to-end model to maximize the likelihood of the target description sentence, given an image, instead of requiring sophisticated data preparation or a pipeline of specifically designed models. Thus we can develop a generative model, a probabilistic framework, based on deep recurrent architecture that combines advances in computer vision and machine translation to generate natural sentences describing an image.

Chapter 1

Introduction

Image captioning is a task that a machine learns to generate natural language sentences to describe the salient parts of an image. Being able to automatically describe the content of an image using properly formed English sentences is a very challenging task. This task is significantly harder, than the well-studied image classification or object recognition tasks, which have been a main focus in the computer vision community. The meaningful description generation process of high level image semantics requires not only the understanding of objects or scene recognition in the image, but also the ability to analyse their states, understand the relationship among them and generate a semantically and syntactically correct sentence. Thus in addition to visual understanding a language model is needed.

The challenge of Automatic image description is to design a model that can fully use image information to generate more human-like rich image descriptions. This task encompasses two kinds of problems: Understanding an Image, which is a Computer Vision(CV) task and Generating a meaningful and grammatically-correct description of the image, which is a Natural-Language Processing(NLP) task. Therefore, to tackle this task it is necessary to advance the research in the two fields, CV and NLP, as well as promoting the cooperation of both communities to address the specific problems arising when combining both tasks.

Generating complete and natural image descriptions automatically has large potential effects in any domain in which images need to be interpreted by humans, but human availability is scarce, or the task at hand is tedious. Such use Cases with text based image retrieval titles attached to news images, descriptions associated with medical images, information accessed for blind users may surely benefit from algorithms able to automatically generate textual image descriptions.Concept-Based Image Retrieval thus has important theoretical and practical research value. Therefore, image captioning is a more complicated but meaningful task in the age of artificial intelligence.

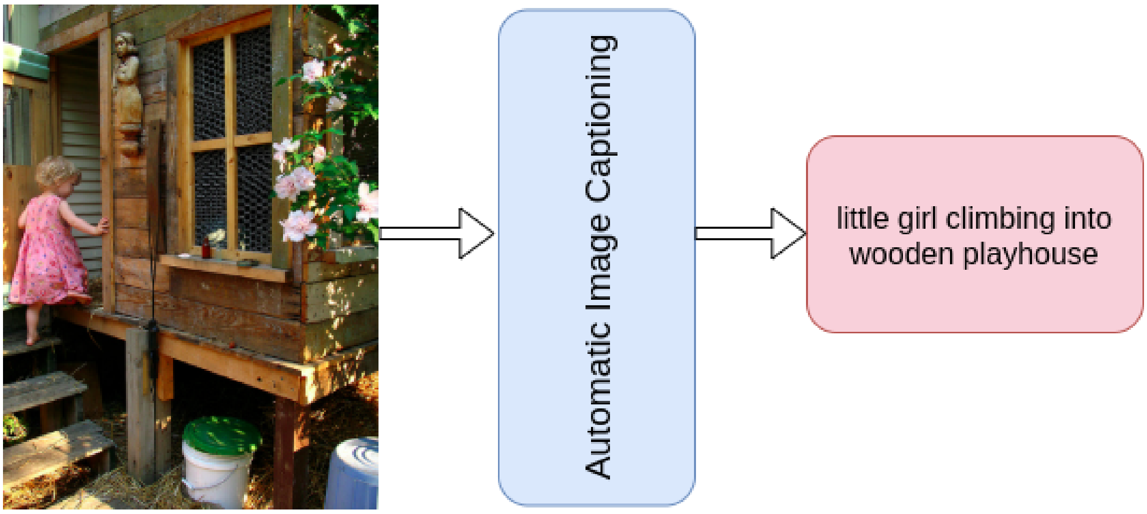

Automatic Image Captioning
little girl climbing into wooden playhouse

Figure 1.1: Automatic Image Captioning

1.1 Objectives

Through this project, we aim to achieve the following objectives

1. Get a solid understanding of the Image Captioning Problem and review the state-of-art solutions to it.
2. Gain in-depth knowledge of Sequence Modeling along with mathematical models of GRU and LSTM.
3. Develop a model on benchmark dataset- Flickr8k with following architecture
   - CNNs to produce a rich representation of the input image by embedding it into a fixed-length vector.
   - RNN network that obtains historical information through the continuous circulation of the hidden layer activation that generates a word sequence.
4. Implement Inject and Merge architectures for integrating image features with language model. Understand and exploit the role of RNN as generator and encoder in given architectures respectively.

Chapter 2

Literature Review

2.1 Methods not based on Deep Learning

In traditional machine learning, hand crafted features such as Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), the Histogram of Oriented Gradients (HOG), and a combination of such features are widely used. In these techniques, features are extracted from input data. They are then passed to a classifier such as Support Vector Machines (SVM) in order to classify an object. The approaches not based on Deep Learning are categorized into two groups : Retrieval based approaches and Template-based approaches.

2.1.1 Retrieval Based Approach

Early work on the topic was often based on the use of retrieval-based approaches, also referred to as transfer-based approaches. These approaches usually follow a two steps process.

1. Given a query image, a candidate set of similar images is retrieved using content-based image retrieval techniques, which are based on global image features extracted from the image.

2. Re-ranking of the retrieved images is computed using a variety of methods. Finally, the caption of the top image is returned, or a new caption is composed of the captions of the top-n ranked images.

The first significant work in solving image captioning tasks was done by **Farhadi et al. (2010)** where three spaces are defined namely the image space, meaning space and the sentence space where mapping is done from the respective image and sentence space to the meaning space. With the help of mapping, similarity between the images and the sentence is evaluated, the meanings are stored as triplets of (image, action, object) and a score is evaluated by predicting the image and sentence triplets. If an image and sentence have high level of similarity in terms of the predicted triplets then they will be highly compatible and have a high score. Thus, appropriate sentences can be generated. This model has many drawbacks such as requirement of the middle meaning space and the results obtained from it are not at all highly accurate.

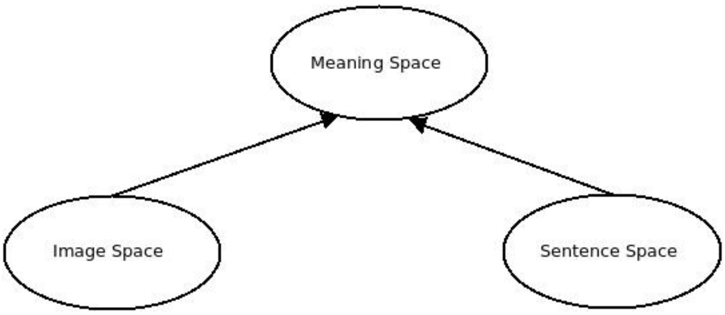


Figure 2.1: Every Picture Tells a Story: Generating Sentences from Images

The IM2TEXT model by **Ordonez et al. (2011)** uses the scene-centered descriptors of the GIST model **(Oliva and Torralba, 2006; Torralba et al. (2008)** to retrieve a set of similar images as a baseline, then these images are ranked using a classifier trained over a range of object detectors and scene classifiers specific to the entities mentioned in the candidate descriptions. Finally, the caption of the top-ranked image is returned. This work was adapted by **Ordonez et al. (2011)** with modifications. Instead of performing a single retrieval step to get neighbors of the query image, this model carries out a separate retrieval step for each visual entity detected in the query image by IM2TEXT. As a result, a collection of different noun, verb and prepositional phrases is retrieved. Using constraint optimization approach a new sentence is composed of selected set of phrases.

The approach adopted by **Gupta et al. (2012)** segments the textual descriptions of the retrieved set of similar images to obtain different types of phrases. The model takes the phrases associated to the retrieved images, rank them based on image similarity, and integrate them to get triples of the form (((attribute1, object1), verb),(verb, prep, (attribute2, object2)), (object1, prep, object2)). Finally, the top-n triplets are used to generate an image description.

2.1.2 Template Based Approach

The Template Based approach analyzes the query image and uses the template as a constraint mechanism to compose a sentence. The template have a number of blank slots and generates sentences with the same syntactical structure.

Kulkarni et al. (2011) employs probabilistic modelling known as Conditional Random Field (CRF) to model the relationships between the image content. Detectors are used to detect the candidate objects such as person, bus, car, trees etc referred to as objects and stuff which are further processed by attribute classifiers. These candidate regions are processed by prepositional relationship function to infer the spatial relatioship between the objects. The Objects and stuff, prepositions and attributes for the nodes while pairwise potential functions are obtained on a collection of existing descriptions. CRF inference is used to determine the image contents to be described, and finally, a sentence template is applied to generate a sentence using the selected content.

Yang et al. (2011) uses $\langle Noun - Verb - Scene - Preposition \rangle$ as a template. Detection algorithms are used to predict the objects and scene. Thereafter the nouns,verbs and preposition for this template are extracted by a Language Model trained on a Gigaword corpus. Hidden Markov Model Inference is used to compute probabilities for all the elements. The elements with the highest highest log-likelihood ratio is used to fill the template and generate a new sentence.

2.2 Deep Learning based Approaches

Encouraged by the success of CNN to solve image classification tasks, researchers began incorporating deep models into their image captioning methods, yet still influenced by the retrieval-based and template-based methods. Deep machine learning based techniques learn features automatically from training data and they can handle a large and diverse set of images and videos.

2.2.1 Multi Modal Learning

The multimodal space part maps the image features into a common space with the word features. The model predicates new words based on the image feature and previously generated context words.

An initial work in this area proposed by **Kiros et al. (2014a)** which uses a CNN for extracting image features using a multimodal space that jointly represents image and text. It also introduces the multimodal neural language models such as Modality-Biased Log-Bilinear Model (MLBL-B) and the Factored 3-way Log-Bilinear Model (MLBL-F). Unlike most previous approaches, this method does not rely on any additional templates, structures, or constraints. Instead it depends on the high level image features and word representations learned from deep neural networks and multimodal neural language models respectively.

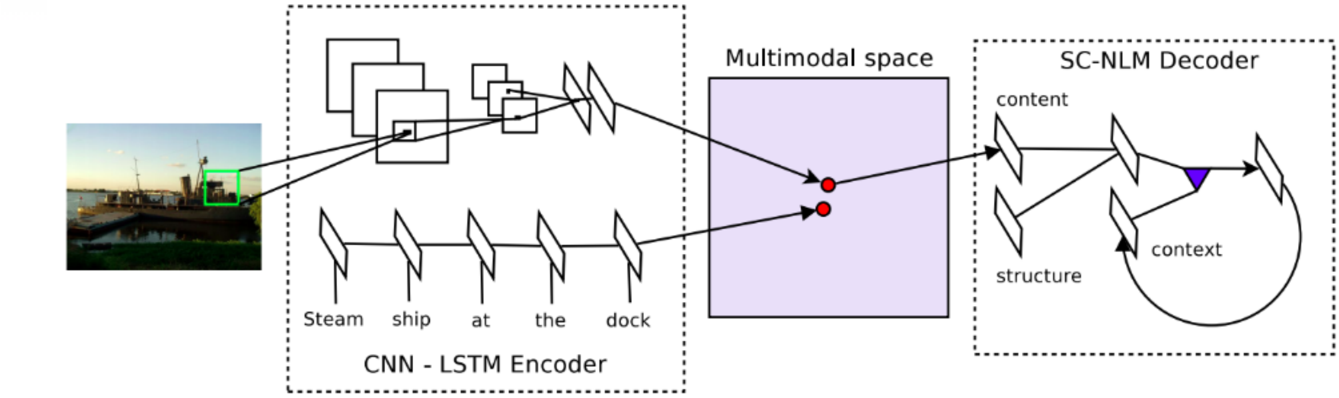


Figure 2.2: Multimodal Learning : Mapping Image and Text to same embedding space

Kiros et al. (2014b) is an extention of this work in to learn a joint image sentence embedding where CNN is used to encode visual data, LSTM is used for sentence encoding and a new neural language model called the structure-content neural language model (SC-NLM) is used for image captions generations. By optimizing a pairwise ranking loss, encoded visual data is projected into an embedding space spanned by LSTM hidden states that encode textual data.

Mao et al. (2014); Mao and Yuille (2015) present a multimodal Recurrent Neural Network (m-RNN) model for generating novel image captions. It directly models the probability distribution of generating a word given previous words and an image and uses this distribution to generate image captions. The model consists of two sub-networks: a deep RNN for sentences and a deep CNN for images. These two sub-networks interact with each other in a multimodal layer to form the whole m-RNN model. Besides generating captions, the m-RNN model can be applied for retrieving images or sentences.

2.2.2 Encoder-Decoder Architecture and Compositional Architecture

Encoder-Decoder Architecture

The neural network-based image captioning methods work as just simple end to end manner. These methods are very similar to the encoder-decoder framework-based neural machine translation **(Sutskever et al., 2014)**. In this network, global image features are extracted from the hidden activation of CNN and then fed them into an LSTM to generate a sequence of words.

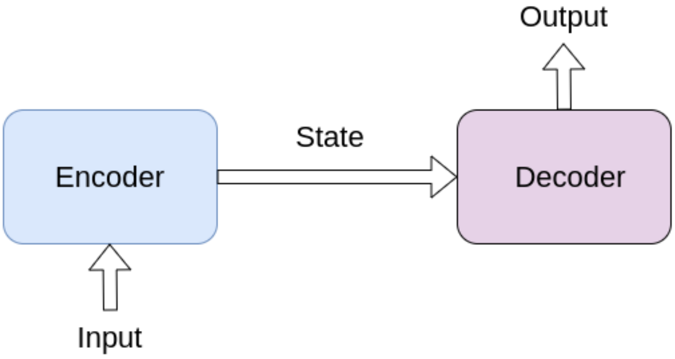


Figure 2.3: Encoder-Decoder: End to End Model

Kiros et al. (2014b) were the first to apply the encoder-decoder framework to generate image descriptions by unifying joint image-text embedding models and multimodal neural language.

Vinyals et al. (2015) proposed a method similar to the one by **Kiros et al. (2014a)** called Neural Image Caption Generator (NIC). It uses CNN for image representations and this image information is passed into the initial state of the LSTM. The LSTM generates next words based on the current time step and the previous hidden state based on maximum likelihood estimation. Since image information is fed only at the beginning of the process, it may face vanishing gradient problems. The role of the words generated at the beginning is also becoming weaker and weaker. Therefore, LSTM is still facing challenges in generating long length sentences.

With an aim to mitigate this problem **Jia et al. (2015)** proposed an extension of the LSTM model, called guided LSTM (gLSTM). This model adds semantic information from the image, and this information is included as an extra input to each gate and cell state of the LSTM network, with the aim of guiding the model towards more relevant solutions. Therefore gLSTM can generate long sentences

Compostional Architecture

Instead of training the parameters for Vision and language by building a tightly coupled system, we can build a compositional architecture connected through a pipeline. The idea behind this architecture is to make modular systems where it is easier to reuse existing components and replace components with little effort.

Fang et al (2015) describes a generation-based system with three modules. It uses visual detectors, a language model, and a multimodal similarity model to train the model on an image captioning dataset. A vocabulary is formed using 1000 most common words from the training captions. The subregion features of an image extracted using CNN are mapped to the words in vocabulary by using Multiple instance learning (MIL) and a Maximum Entropy model is used to generate sentences from these features.

Ma and Han (2016) propose another method using a compositional architecture. Their main contribution is the introduction of structural words; tetrads composed of $\langle objects, attributes, activi$
The method consists of two stages: structural word generation and sentence translation. At the first stage, multi-layer optimization is applied to generate a hierarchy of concepts that will play the role of structural words. During the second stage, an LSTM is used to translate the structural words into full sentences.

2.2.3 Attention Networks

Xu et al. (2015) were the first to introduce an attention-based image captioning method. The method describes the salient contents of an image automatically. The main difference between the attention-based methods with other methods is that they can concentrate on the salient parts of the image and generate the corresponding words at the same time. This method applies two different techniques: stochastic hard attention and deterministic soft attention to generate attentions.

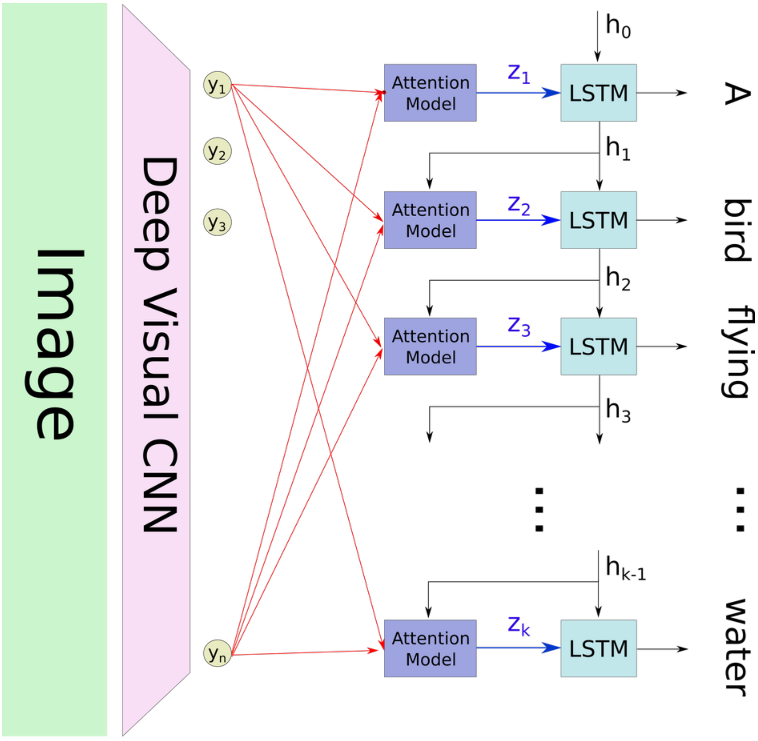


Figure 2.4: Attention Mechanism

A common drawback of these spatial attention methods are that they compute weighted pooling only on attentive feature map. As a result, these methods lose the spatial information gradually. Moreover, they use the spatial information only from the last convolution layer of the CNN. The receptive field regions of this layer are quite large that make the limited gap between the regions. Therefore, they do not get significant spatial attentions for an image. To deal with these limitations of spatial attention, **Chen et al. (2017a)** proposed another attention-based image captioning method that combines spatial attention with channel wise attentions to compute an attention map.

Chapter 3

Problem Statement

For the image captioning task, humans can easily understand the image content and express it in the form of natural language sentences according to specific needs; however, for computers, it requires the integrated use of image processing, computer vision, natural language processing and other major areas of research results. To achieve this,

- For the Vision task, it is necessary to get a good understanding of the scene. In addition to the detection of objects, scene recognition -capturing the relations between the objects and their interaction, is paramount.
- Similarly from NLP point of view, the description should sum up the important elements in the image. The should be comprehensive and concise. For this, the model should select the visual aspects to verbalize and generating a semantically and syntactically correct sentence.

Intuitively, descriptions should be easy to understand by a person, and that person should be able to grasp the essence of the image, to create a mental model of the image without actually seeing it.

Chapter 4

Scope

Develop a single end to end encoder-decoder network to develop more accurate feature extraction and efficiently generate textual description which can provide detailed information about the given image.

- We propose to adapt the elegant receipe of Machine Translation where translation work is achieved by using an encoder RNN that reads the source sentence and transforms it into a rich fixed-length vector representation, which in turn is used as the hidden state of a decoder RNN that generates the target sentence.
- To solve the task of image captioning using encoder-decoder approach we replace the encoder RNN by a deep CNN which can produce a rich representation of input by embedding it in a fixed-length vector.

Apply and learn different views on stages at which the image information can be introduced to the language model. To implement this we leverage novel architectures like InceptionV3 to extract the visual features from an input image. Based on this we condition out language model on following two approaches

- Injecting the Image - The image information is fed to the neural network by directly incorporating it in the RNN.
- Merging the Image - The image features are introduced in a layer following the RNN by the merging it with the output of processed words by RNN.

In language model, decoding the most likely output sequence involves searching through all the possible output sequences based on their likelihood. To return one or more "good" predictions we apply heuristics search methods like Greedy Search and Beam Search for caption generation and study the effect on performance of the Model.
Finally, evaluate generated captions with respect to the reference translations for each of the above cases using Evaluation Metrics like Bleu Score on unigram, bigrams and n-grams.

Chapter 5

Technology Stack

5.1 Platform

- Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

5.2 Libraries

- PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing. It is primarily developed by Facebook's artificial intelligence research group
- NumPy is the fundamental package for scientific computing with Python. It contains among other things
  - a powerful N-dimensional array object
  - sophisticated (broadcasting) functions
  - tools for integrating C/C++ and Fortran code
  - useful linear algebra, Fourier transform, and random number capabilities
- Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
- Keras is an Open Source Neural Network library written in Python and a high-level API wrapper for the low-level API that runs on top of Theano or Tensorflow. Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models.
- Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- OpenCV library for preprocessing images.

Chapter 6

Benefits for the environment

- Generate captions for images which can promote safety and protection of environment
- Determine various pollutants present in a given image and caption them so as to reduce its generation and manage it.
- Image search tools can help in finding environmental problems and its solutions.

Chapter 7

Benefits for the Society

- Helps visually impaired to understand the image by converting the captioned text into speech.
- Helps colour blind and other vision problem patients to understand image more effectively.
- Providing more accurate captions for images which can be easily misunderstood.
- Finding hidden features from keywords generated by captions.

Chapter 8

Application

The web is filled with billions of images, helping to entertain and inform the world on a countless variety of subjects. The existing NLP applications that benefit which extract insights/summary from given text data or an essay etc can be extended to people who would benefit from automated insights from images. While automatic image captioning can help solve this problem, accurate image captioning is a challenging task that requires advancing the state of the art of both computer vision and natural language processing. Elaborate applications in image captioning are :

- Help Visually Impaired:
  Much of the visual information is not accessible to those with visual impairments. A rich automated description of the image would benefit them.
- E -commerce assistant:
  Image captions, manually added by website authors using Alt-text HTML, is one way to make this content more accessible, so that a natural-language description for images that can be presented using text-to-speech systems. However, existing human-curated Alt-text HTML fields are added for only a very small fraction of web images.
  - Mapping images to natural language and vice versa would also help in medical image understanding that conveys the clinical physician that the algorithm has found something fishy in the patient by mapping the physiological parameters and images which may require further investigations.
  - Platforms like Facebook can infer directly from the image, where you are (beach, cafe etc), what you wear (colour) and more importantly what you're doing also (in a way).
  - These applications can be extended to explaining what is happening in a video frame by frame.

Chapter 9

Dataset Selection and Statistics

9.1 Datasets Available for Caption Generation Task

A large amount of work has been done on image caption generation task. Most of the significant work in solving caption generation tasks involve following benchmark datasets.

Pascal1k

Each image is this dataset is associated with 5 captions per images and these are collected via crowdsourcing. By associating each image with five captions, it captures the linguistic variety in describing an image.

Flickr8k

Flickr8k uses a strategy similar to Pascal Flickr8k is a novel dataset for sentence description of an image with 8000 images obtained from the Flickr photo-sharing website. The dataset covers wide variety of images with everyday actions and events featuring images and actions.

Flickr30k

Flickr30k is an extension of Flickr8k dataset and is a standard benchmark for sentence-based image description. The dataset contains manually annotated bounding boxes for each image.The dataset mainly consists of humans in everyday activities and is complemented with a denotation graph that pairs generalized versions of the image captions with their visual denotations (the sets of images they describe)

Microsoft COCO (Common Objects in Context)

MSCOCO is a large-scale object detection, segmentation, and captioning dataset. The dataset provides 5 captions per image for image captioning. It includes 123287 color images (10^6 magnitude).The images in MSCOCO have been annotated with 5 descriptions per image, plus bounding boxes for objects belonging to 80 object categories.MSCOCO Dataset consists of images with complex everyday scenes of common objects in their natural context. This dataset focuses on scene understanding along with object Detection

Google's Conceptual Caption

- Google's Conceptual Captions dataset has more than 3 million images, paired with natural-language captions. Conceptual Captions images and their raw descriptions are harvested from the web, and therefore represent a wider variety of styles. The raw descriptions are harvested from the Alt-text HTML attribute associated with web images. An automatic pipeline extracts, filters, and transforms candidate ¡image, caption pairs¿, with the goal of achieving a balance of cleanliness, informativeness, fluency, and learnability of the resulting captions. This pipeline is known as Flume pipeline which processes billions of web pages parallely.

The Datasets can be summarized as follows :

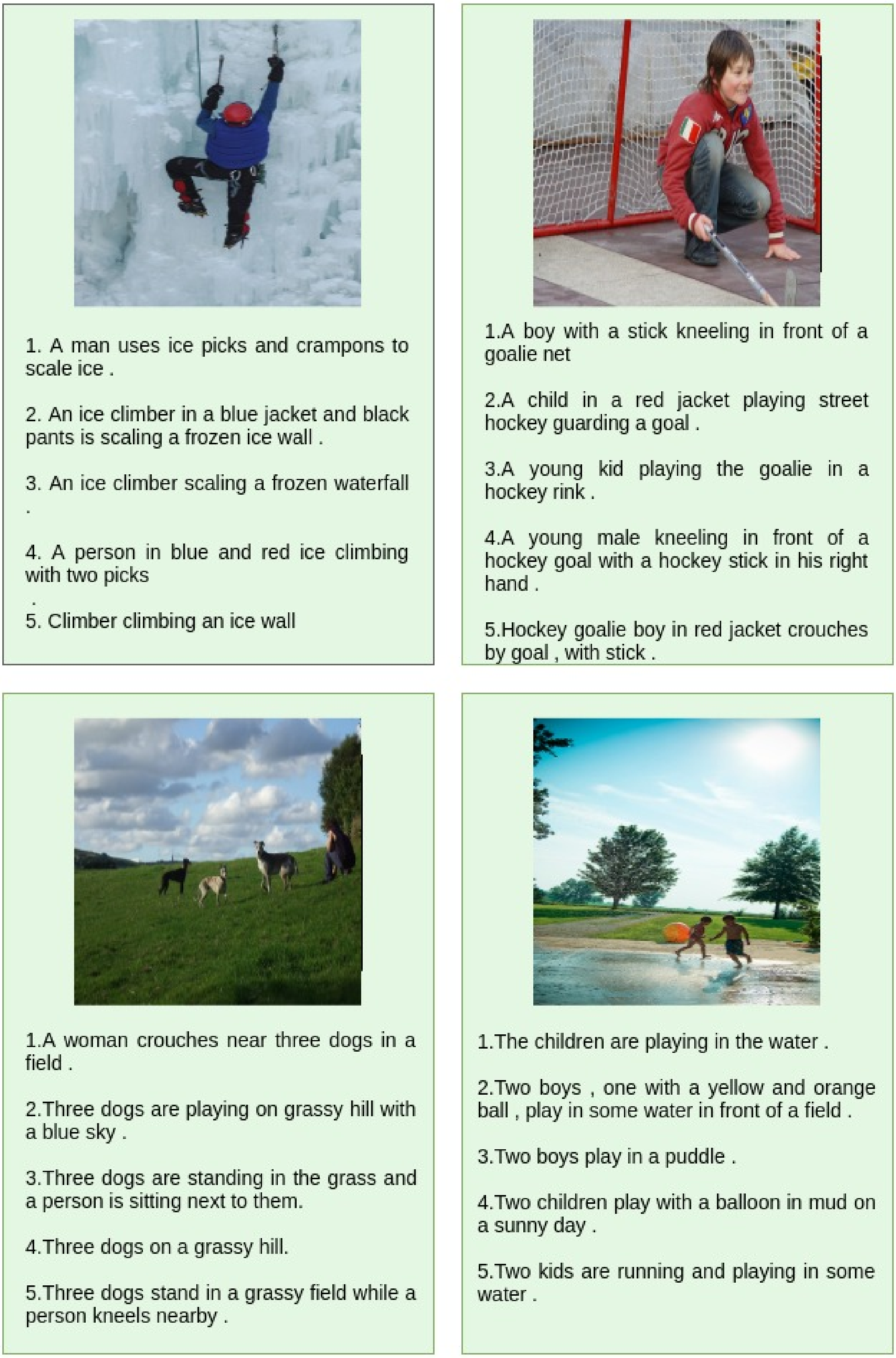
Dataset	Paper	Size			Caption per
—	Name	Train	Valid	Test	Image
Pascal1K	(Rashtchian et al., 2010)	—	—	1000	5
FLickr 8K	(Rashtchian et al., 2010)	6000	1000	1000	5
FLickr 30K	(Young et al., 2014)	28000	1000	1000	5
MSCOCO	(Lin et al., 2014)	82783	40504	40775	5
Conceptual Captioning	(Sharma et al., 2018)	318333	28355	28530	1

Table 9.1: Datasets for Caption Generation

9.2 Understanding the Data

For the purpose of our study, we use Flickr8k. Flickr8k is realistic and relatively small. We can download it and build models on our workstation using a CPU. Flickr8k dataset is available for free and contains

- Flicker8k Dataset: Contains a total of 8092 images in JPEG format with different shapes and sizes.
  - 6000 — Training
  - 1000 — Validation
  - 1000 — Testing
- Flickr8k text: Contains text files describing trainset ,testset. Flickr8k.token.txt contains 5 captions for each image i.e. total 40460 captions.



1. A man uses ice picks and crampons to scale ice .

2. An ice climber in a blue jacket and black pants is scaling a frozen ice wall .

3. An ice climber scaling a frozen waterfall .

4. A person in blue and red ice climbing with two picks .

5. Climber climbing an ice wall

1.A boy with a stick kneeling in front of a goalie net

2.A child in a red jacket playing street hockey guarding a goal .

3.A young kid playing the goalie in a hockey rink .

4.A young male kneeling in front of a hockey goal with a hockey stick in his right hand .

5.Hockey goalie boy in red jacket crouches by goal , with stick .

1.A woman crouches near three dogs in a field .

2.Three dogs are playing on grassy hill with a blue sky .

3.Three dogs are standing in the grass and a person is sitting next to them.

4.Three dogs on a grassy hill.

5.Three dogs stand in a grassy field while a person kneels nearby .

1.The children are playing in the water .

2.Two boys , one with a yellow and orange ball , play in some water in front of a field .

3.Two boys play in a puddle .

4.Two children play with a balloon in mud on a sunny day .

5.Two kids are running and playing in some water .

Figure 9.1: Dataset Visualisation

9.3 Data Cleaning

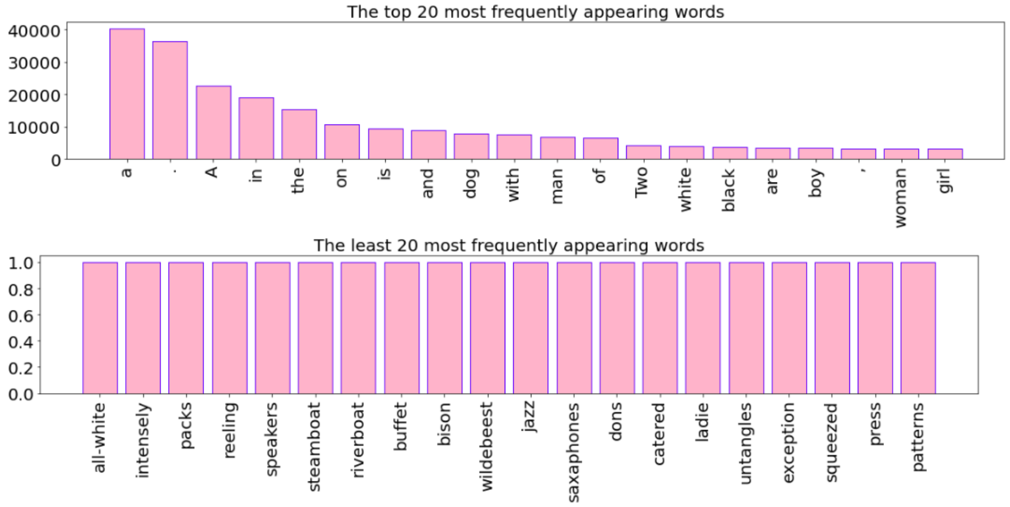


Figure 9.2: Before Preprocessing Captions

Captions are sentences in Natural Language. These Captions need to be cleaned before using it to train the model. We perform the following steps for cleaning the text data.

- Change the caption text to lowercase
- Remove all punctuation from tokens.
- Remove hanging letters 's' and other letters earlier specified with apostrophe.
- Remove extra white spaces.

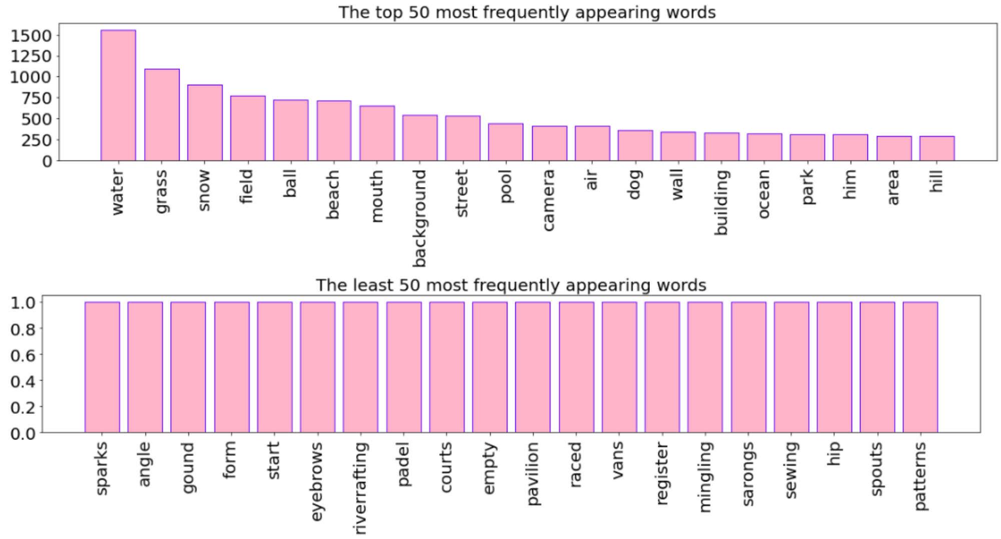


Figure 9.3: After Preprocessing Captions

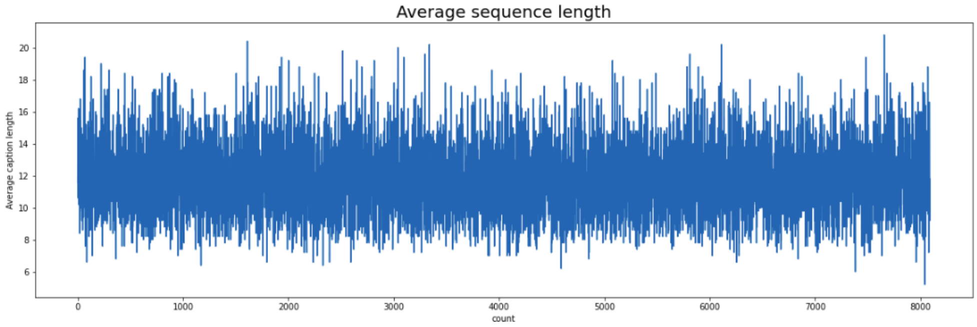


Figure 9.4: Average Length of Caption per Image

Mean	11.7822
Standard Deviation	2.1292
25%	10.4
50%	11.6
75%	13.2
Min	5.2
Max	20.8

Table 9.2: Statistics : Average Length Per Image

Average Length of Caption Per Image

The Maximum Length of the Caption Found is : 34

Chapter 10

Model Architecture

10.1 Overview of the Architecture

The proposed system consists of the following components

- Convolutional Neural Network (CNN) - CNN extracts a feature map for the given input image. From a learning perspective we define a CNN model and perform an end-to-end training on our dataset. Progressing we use InceptionV3- a pre-trained deep learning model which attains an accuracy greater than greater than 78.1% on ImageNet Dataset.

- Recurrent Neural Network (RNN) - The output of the CNN network if forwarded as input to the Recurrent layer for which we use Long Short Term Memory units (LSTM). LSTM produces a caption for the input image, using information from the image embeddings and its hidden state inputs.

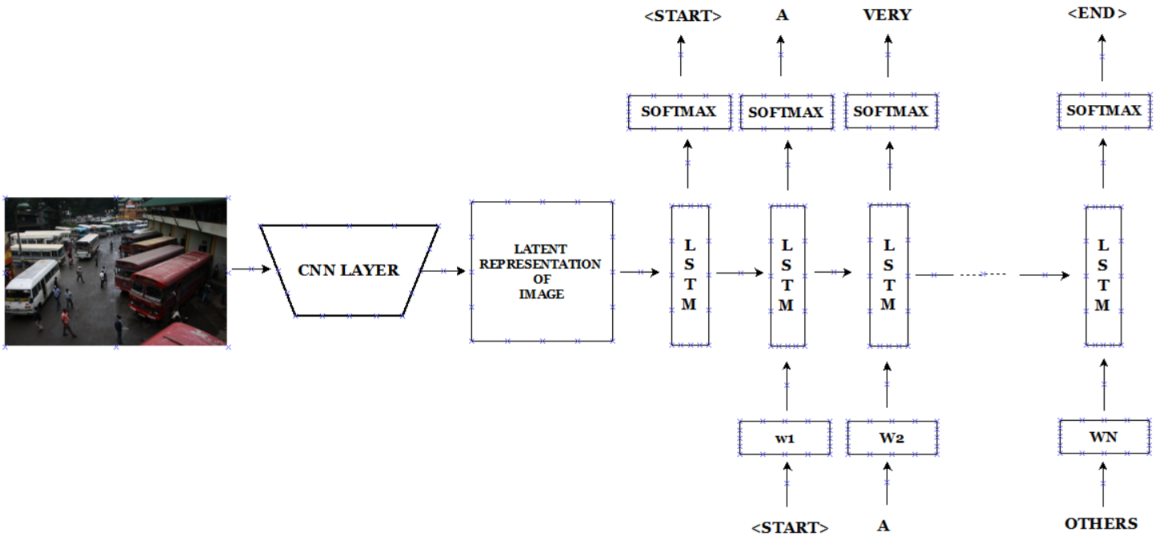


Figure 10.1: Flow Diagram

10.2 Convolution Neural Networks

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which will take in an input image, assign learnable weights and biases to various aspects or objects in the image. The pre-processing required in a Convolutional Neural Net is much lower as compared to other classification algorithms and they have the ability to learn these characteristics of an image. A convolution neural network will produce a rich representation of input image by encoding it to fixed size vectors.The important components of the CNN include

10.2.1 Components of CNN

1. The Convolution Layer

In this layer, the filters (weight matrix) is applied to the input image which extracts certain features such as edges,colors and shapes from the image. The depth dimension of the weight matrix would be same as the depth dimension of the input image as it extends to the entire depth of the input image. Therefore, convolution with a single weight matrix would result into a convolved output with a single depth dimension. The output from the each filter is stacked together forming the depth dimension of the convolved image.

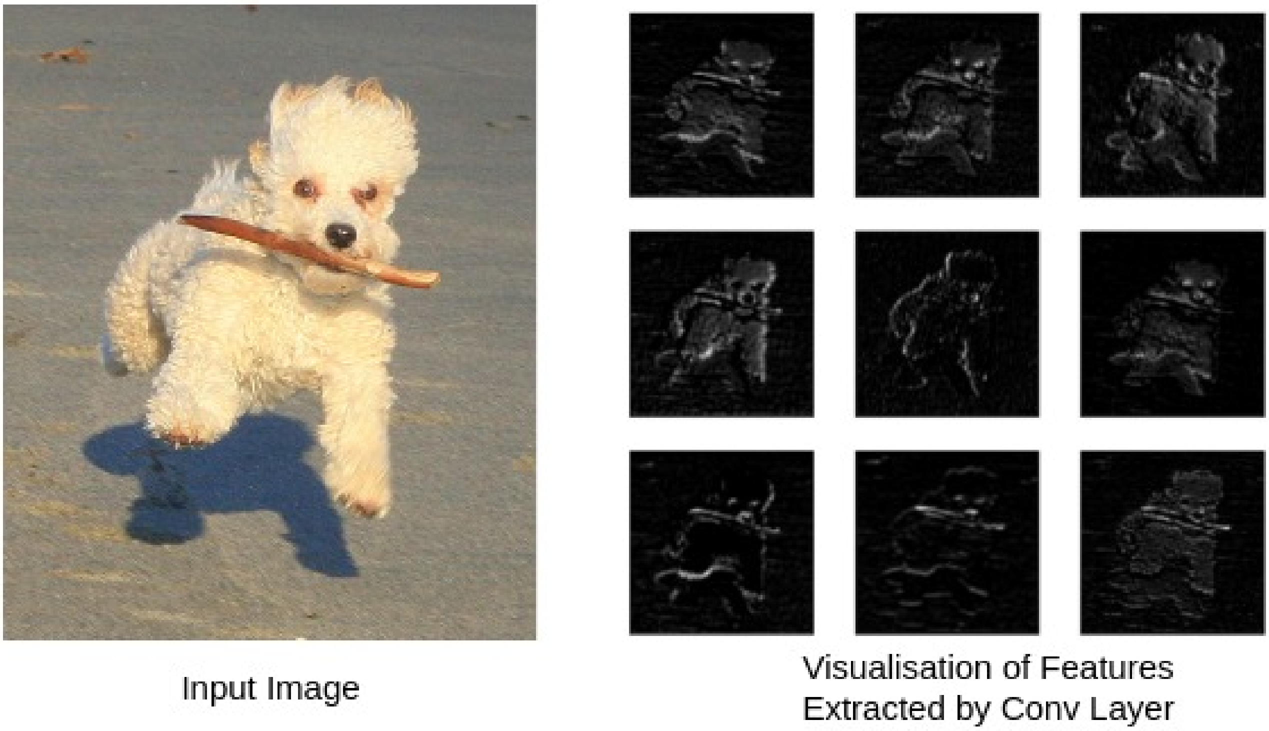


Figure 10.2: Feature Extraction by CNN

2. The Pooling Layer

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction.Convolutional networks may include local or global pooling layers to streamline the underlying computation. There are two types of Pooling:

- Max Pooling - Returns the maximum value from the portion of the image covered by the Kernel.
- Average Pooling - Returns the average of all the values from the portion of the image covered by the Kernel.

Strides and Padding

Stride - Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.
Padding - The image shrinks with the addition of more hidden layers and large strides. Also, the pixels at corners and edges are touched less by the filter in comparison to the pixels at the center. To avoid this problem, padding- a layer of 0s (zeros) is added surrounding the query image.

- Valid Padding - If no padding is applied.
- Same Padding - If we add padding 'p' such that the size of the image is retained then it is called same padding

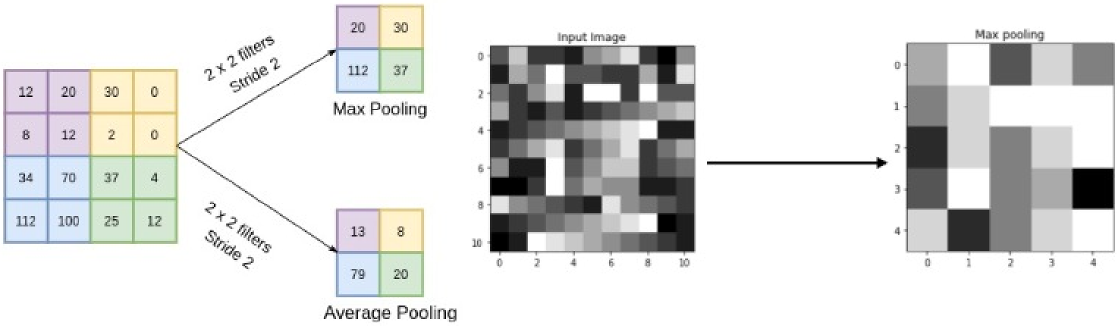


Figure 10.3: pooling layer in CNN

3. Activation Layer

The activation function is commonly a ReLU layer. ReLU (Rectified Linear Units) is non-linear function defined by ReLU(x)=max(0,x). It can be visualised as follows

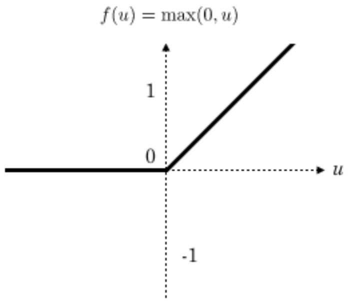


Figure 10.4: ReLu Activation

4. Fully Connected Layer

The convolution and pooling layers extract features and reduce the number of parameters from the original images to generate a 3D feature map. Finally, we apply a fully connected layer to generate an output equal to the number of classes we need.
The Spatial Size of the Image can be calculated as $(\frac{[W-F+2P]}{S}+1) * (\frac{[W-F+2P]}{S}+1)$
where,

W ⇒ Input Volume Size
F ⇒ Size of the Filter
P ⇒ Number of Padding
S ⇒ Number of Strides

10.2.2 InceptionV3

The CNN module consists of pretrained on the ImageNet dataset. Several options are available for this model

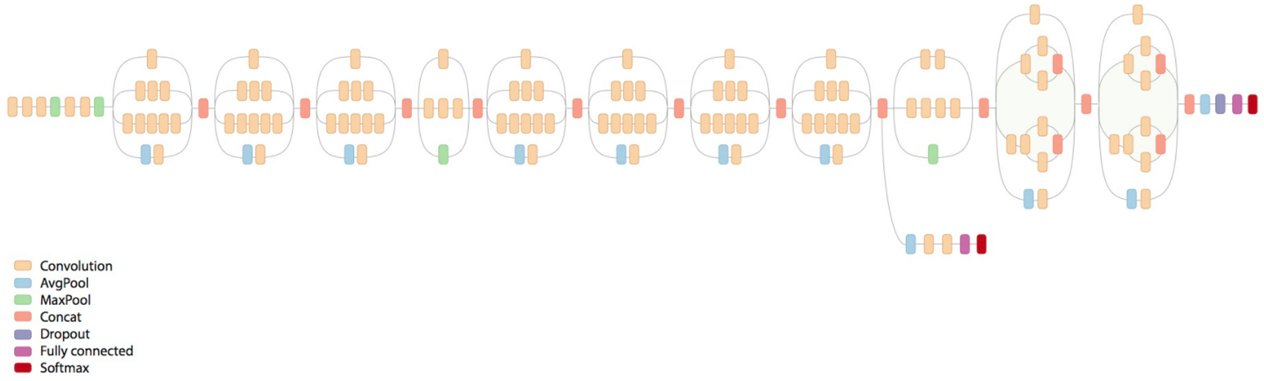


Figure 10.5: InceptionV3 Architecture

Downsampling Feature Maps with 1*1 Convolution

Deep neural networks are computationally expensive. A large number of feature maps in a convolutional neural network can cause a problem as a convolutional operation must be performed down through the depth of the input. To make it cheaper, Inception Network contains 1×1 Convolution at the middle of the network. It limits the number of input channels by adding an extra 1x1 convolution before the 3x3 and 5x5 convolutions.

A 1×1 convolutional layer offers a channel-wise pooling, often called feature map pooling or a projection layer. Though adding an extra operation may seem counter-intuitive, 1x1 convolutions are far more cheaper than 5x5 convolutions, and the reduced number of input channels also help. However, the 1x1 convolution is introduced after the max pooling layer, rather than before. This method was popularized in the paper **Network in Network** by **Min Lin, et al**

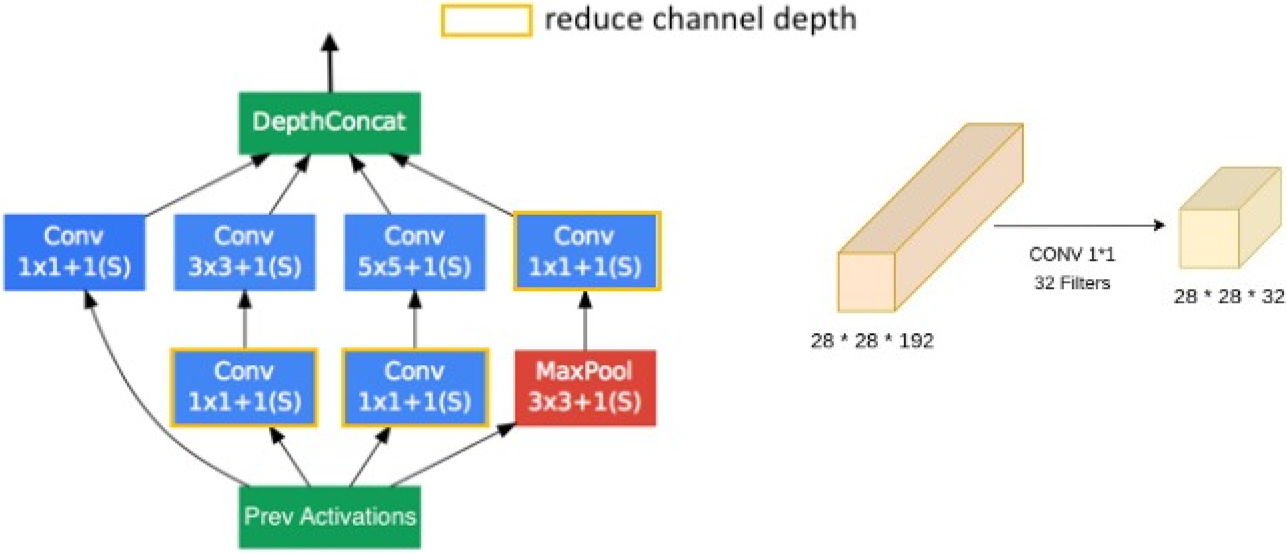


Figure 10.6: InceptionV3 Building Blocks

10.3 Recurrent Neural Networks

The target variable is the captions that our model is learning to predict. The output of the previous module (CNN) is fed as an input to the decoder RNN which would generate the sentences. A Recurrent Neural Network (RNN) can be thought as multiple copies of same network, each passing a message to its successor. Therefore unlike other neural networks where the inputs are independent of each other, inputs in RNN are related to each other. This makes RNN applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

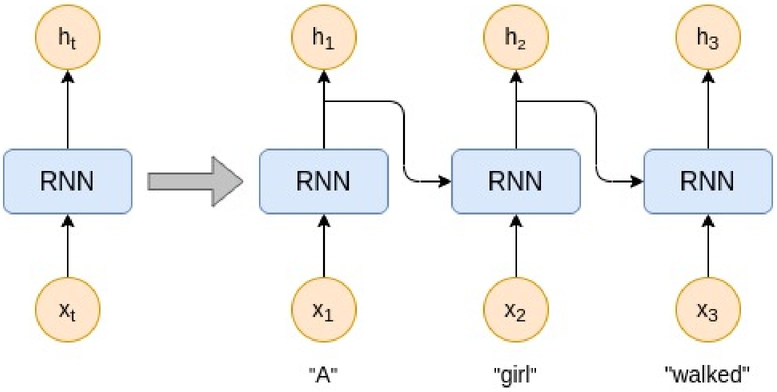


Figure 10.7: Recurrent Neural Network

$$\log p\ (S|A)\ =\ \sum_{t=1}^{N} \log p\ (S_t|I, S_0, ..., s_t - 1) \tag{10.1}$$

where θ are the parameters of our model, I is an encoded image representation, and S its correct transcription. Since S represents any sentence, its length is unbounded. Thus, it is common to apply the chain rule to model the joint probability over S0 ; . . . ; SN , where N is the length of this particular example as

$$\Theta^*\ = \arg\max_{\theta} \sum_{I,S} \log p\ (S|I : \Theta) \tag{10.2}$$

10.3.1 Long Short Term Memory (LSTM)

Traditional Neural Networks do not preserve the information learned in previous times. RNN make use of the sequential information. RNN faces following issues

- Long Term Dependency
- Vanishing Gradient and Exploding Gradient

To model this probability we use Long Short Term Memory (LSTM) which is a special type of RNN. The heart of LSTM is it's cell which provides memory. The cell is made up of three types of gates

1. **Input Gate** : Feeds the new information that we're going to store in the cell state.

2. **Output Gate** : Provides the activation to the final output of the lstm block at timestamp ‘t’.

3. **Forget Gate** : Decides what information to throw away from the cell state.

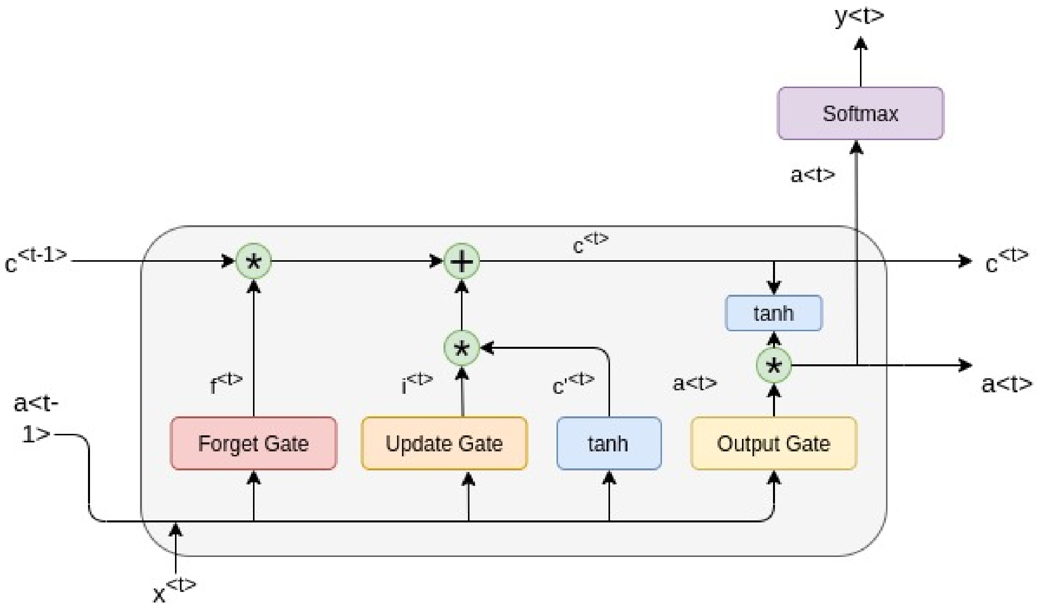


Figure 10.8: Architecture of the LSTM cell

The equations of the LSTM gates are :

$i_t = \sigma(w_i[a_{t-1}, x_t] + b_t)$ $f_t = \sigma(w_f[a_{t-1}, x_t] + b_f)$ $o_t = \sigma(w_o[a_{t-1}, x_t] + b_o)$

where,

$a_t \Rightarrow$ represents input gate
$f_t \Rightarrow$ represents forget gate
$o_t \Rightarrow$ represents output gate
$\sigma \Rightarrow$ represents sigmoid function

$w_x \Rightarrow$ weight for the respective gate(x) neurons
$a_{t-1} \Rightarrow$ weight for previous lstm block(at time t-1)
$x_t \Rightarrow$ input at current timestamp
$b_x \Rightarrow$ Biases for the respective gates(x)

The equations for the cell state, candidate cell state and the final output are:

$\tilde{ct} = \tanh(w_c[h_{t-1}, x_t] + b_c)$
$c_t = f_t * c_{t-1} + i_t * \tilde{ct}$
$h_t = o_t * \tanh(c^t)$

$c_t \Rightarrow$ cell state(memory) at timestamp(t)
$\tilde{ct} \Rightarrow$ represents candidate for cell state at timestamp(t)

*Note : * represents the element wise multiplication of the vectors.*
Lastly, we filter the cell state and then it is passed through the activation function which predicts what portion should appear as the output of current LSTM unit at timestamp t.We

can pass this ht the output from current lstm block through the softmax layer to get the predicted output(yt) from the current block.

10.4 Types of Architecture : Based on CNN-RNN Interaction

RNNs are typically viewed as 'generators' as suggested in Bernardi et al. (2016), LeCun et al. (2015),(Sutskever et al., 2011; Graves, 2013). RNN is trained to generate the next word [of a caption]'.The paper Marc Tanti et al. (2017) provides an alternative approach where the role of the RNN can be thought of as primarily to to encode sequences, but not directly to generate them.

10.4.1 Inject Architecture

In Inject Architectures, the activations derived at the last hidden layer of the CNN are fed to the RNN as the input. This approach involves early inclusion of the image features and takes these features as an caption-prefix input. This language-image mixture is encoded by the RNN to develop an encoding that incorporates both visual and linguistic information together. Thus in inject architecture the features of the image are updated by the RNN on every timestep to its internal representation.

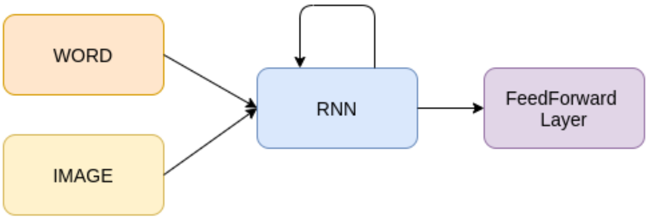


Figure 10.9: Inject Architecture

We can narrow down these architectures further in the following way.

- Init-inject
  The image representation is used as an initial hidden state input to the RNN. The image representation vector is required to have the same size as that of the hidden state vector of the RNN with word vectors as an input.

- Pre-inject
  This configuration is a slight variation of the Init-inject method. Instead of giving the image representation as an initial hidden state input, it is fed as a first input to RNN followed by word vectors. The image vector is of the same size as that of word vector size.

- Par-inject
  In this approach, along with word-vectors, the image vectors also act as an input to the RNN. The RNN may take two separate inputs or the combined image-text vectors may serve as an input.

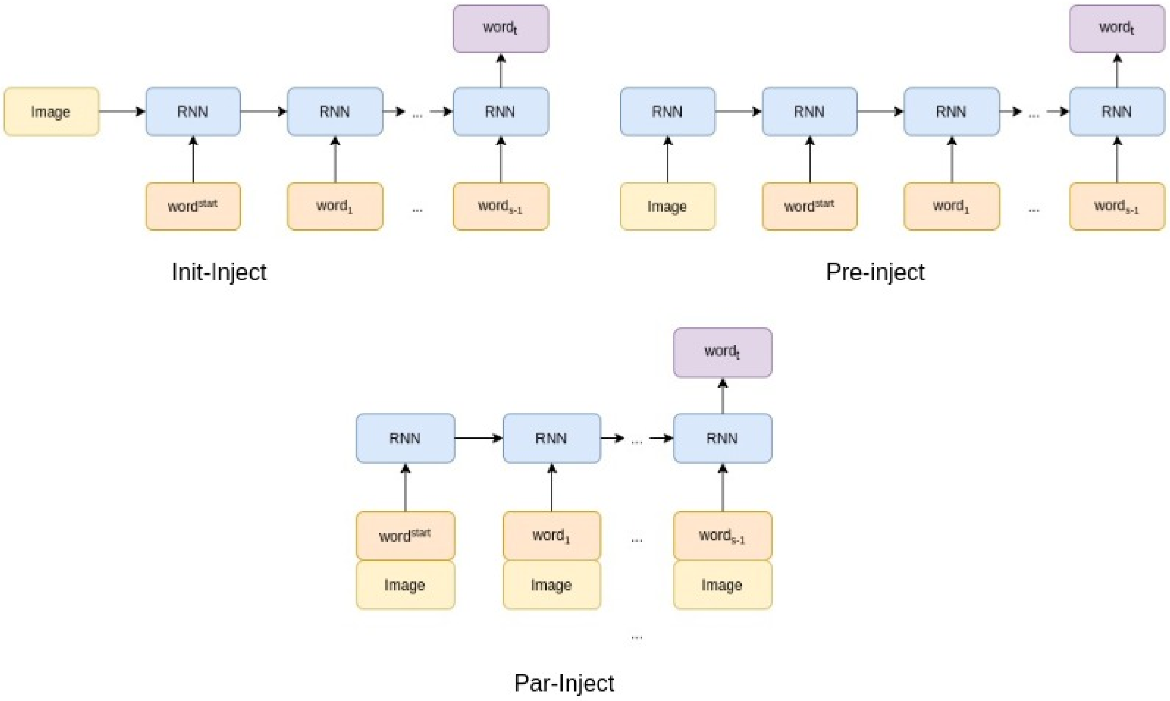


Figure 10.10: Types of Inject Architectures

10.4.2 Merge Architecture

Merge Architectures tend to incorporate image features late into the text generation process as shown below. In these architectures, RNN Networks handle purely the linguistic information. The image is left out of the RNN network. Unlike Inject architecture, fixed image representation is used for the prediction of word at every time step. The image embeddings and text description generated so far are combined. This combination are fed to the feed-forward layer for next word prediction.

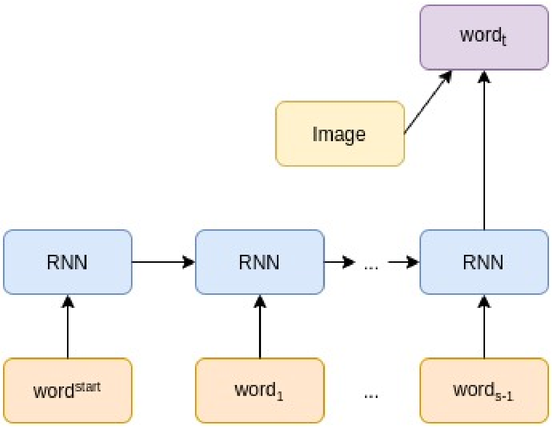


Figure 10.11: Merge Architecture

Chapter 11

Data Pipeline

11.1 Image Preprocessing

To begin with, we have to encode the images by CNN to extract the image embeddings. The pre-processing required in a Convolutional Neural Net is much lower as compared to other classification algorithms. However we may need to perform basic pre-processing such as resizing them to the shape required by the particular CNN being used, as well as applying tweaks that are specific of each particular CNN. Some models use images with values ranging from 0 to 1,others from -1 to +1. We can encode the images and save the extracted features of our dataset images before training and inference. This save a lot of processing overhead.

Keras ships state-of-art CNNs pre-trained on ImageNet dataset. The characteristics such as required input size, number of parameters, weights size of these models can be summarized as:

Model	**Image Size**	**Weights Size**	**Top-1 acc.**	**Top-5 acc.**	**Params**	**Depth**
Xception	299 x 299	88 MB	0.790	0.945	22,910,480	126
VCG16	224 x 224	528 MB	0.715	0.901	138,357,544	23
VCG19	224 x 224	549 MB	0.727	0.910	143,667,240	26
ResNet50	224 x 224	99 MB	0.759	0.929	25,636,712	168
InceptionV3	299 x 299	92 MB	0.788	0.944	23,851,784	159

Table 11.1: Novel CNN Architectures

We will be using InceptionV3 as our CNN module. Since the weights size is less it is faster to train. We reshape our input to (299,299) and normalize the pixel values. The encoding of the image obtained is obtained by the last layer of the network, excluding the fully connected part pre-trained for classification task. Depending on the architecture, the size of the feature vector may differ. We pop the last layer of the InceptionV3 network to obtain a feature vector of size (1,2048) for every training image.

11.2 Text Preprocessing

To implement Teacher Forcing, we split each of the training sequences into ⟨ training-input, training-output ⟩ pairs:
Consider the caption : ”girl going into wooden building”

- Input (X) : The Input Sequence is append the start of sentence token ’⟨ sos ⟩’. Therefore the Training-Input Sequence becomes ”⟨ sos ⟩ girl going into wooden building”

- Output (Y) : The Output for the given sequence is one step ahead of the input sequence for the same training example. We append end of sequence token ”⟨ eos ⟩” to the sequence. Therefore the Training-Output Sequence becomes ”girl going into wooden building ”⟨ eos ⟩”

Based on the Image encoding generated, model generates captions by predicting next words. We cannot work directly with the text. Therefore, we need to map the captions to numerical vectors.
To do this, an internal vocabulary(dictionary of words) from the list of texts must be constructed. This vocabulary should be constructed based on the word frequency. Now each caption can be converted to sequence of integers based on this dictionary of words. So it basically takes each word in the text and replaces it with its corresponding integer value from the word-to-index dictionary. Lastly, we pad these sequence of numbers to maximum length of the caption in the dataset. This provides the fixed number of time steps to our language model.
To accomplish this task, we used the sophisticated API provided by Keras that can be fit on multiple documents—Tokenizer. Fitting list of texts on this Tokenizer objects updates internal vocabulary based on a list of texts. Lower integer means more frequent word. 0 is reserved for padding.

11.3 Data Generator

One way to Data Generator are used to generate data in batches rather than giving the entire data at once. Now we organize our preprocessed image and text data to special data structures supporting batching.

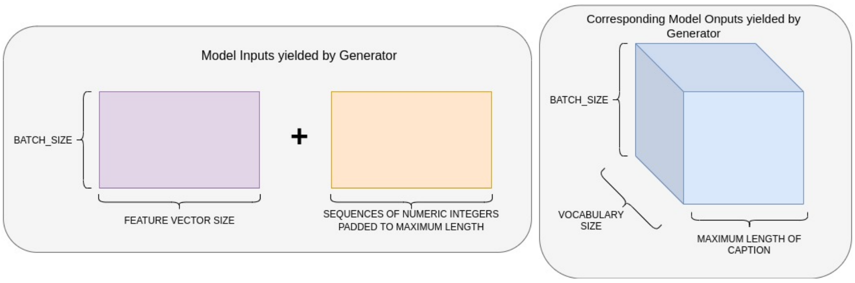

Model Inputs yielded by Generator
BATCH_SIZE
+
FEATURE VECTOR SIZE
SEQUENCES OF NUMERIC INTEGERS PADDED TO MAXIMUM LENGTH
Corresponding Model Onputs yielded by Generator
BATCH_SIZE
VOCABULARY SIZE
MAXIMUM LENGTH OF CAPTION

Figure 11.1: Data Structures for Model Training

Chapter 12

Training

12.1 Teacher Forcing

Sequence Prediction Models often use output from the last time step y(t-1) as input for the model at the current time step X(t). However during the early stages of learning, the hidden states of the model are often updated by a sequence of wrong predictions. Feeding this as an input to the current timestep causes errors to accumulate, and it is difficult for the model to learn from that.

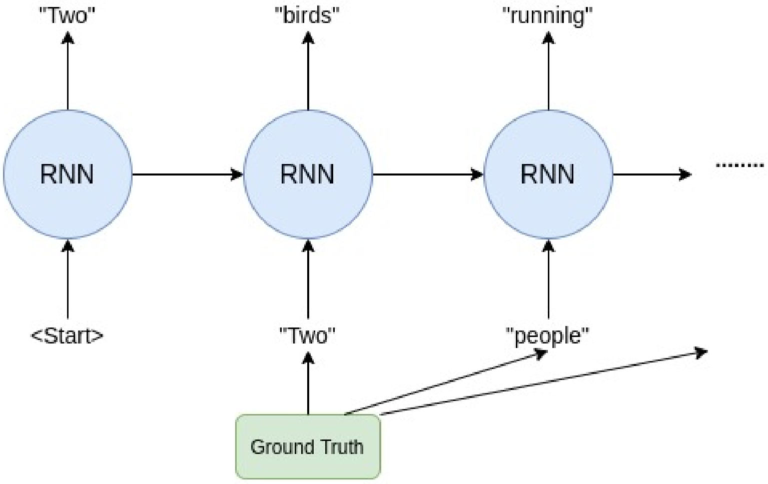


Figure 12.1: Without Teacher Forcing

Teacher Forcing remedies this by using the output in the prior time step(available in the training data) to compute the system state in the current time step. Instead of summing activations from incoming units, each unit sums correct teacher activations as input for the next iteration. It is a procedure for training RNNs with output to hidden recurrence which emerges from maximum likelihood criterion. Thus the model converges faster.

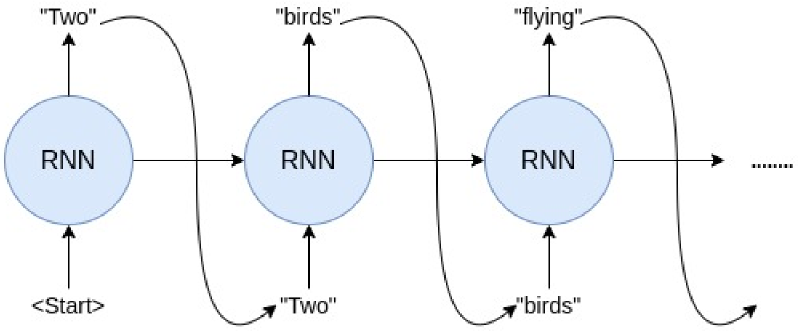


Figure 12.2: With Teacher Forcing

Exposure Bias

During inference the RNN feed its own previous prediction back to itself for the next prediction since ground truth is not available. This divergence between training and inference might lead to poor model performance and instability. This is known as Exposure Bias in literature.

Chapter 13

Inference

After the model is trained, it can be used to generate new sequences. To do that, we should simply feed a ”⟨ sos⟨ ” token to the RNN, and the generated word in the output sequence will then be used as input on the subsequent time step, along with the context vector coming from the attention mechanism. The generation process will continue until the ”⟨ eos ⟩ ” token is reached, or a maximum sequence length is reached. Following Approaches are used for prediction of next word

13.1 Greedy Search

Greedy search selects the most likely word at each step in the output sequence. Here we select the word that has the highest probability (i.e act greedily). Choosing just one best candidate might be suitable for the current time step, but when we construct the full sentence, it may be a sub-optimal choice.

13.2 Beam Search

Beam search has a hyperparameter 'B' known as Beam Width. Depending on this hyperparameter, Beam search considers number of multiple alternatives. At every time step, it selects B best alternatives with the highest probability as the most likely possible choices.
The beam width balances between the quality of the description generated and computational overhead. A lower beam width will result in more inferior quality translation but will be fast and efficient in terms of memory usage and computational power whereas a higher beam width will use lot of memory and computation power but gives the optimal description. We implement beam search on our merge-model we will note the results for beam width 3,5 and 7.

Suppose we have beam width as 3. Now we want to generate the best caption using beam search. Here the algorithm takes input as start sequence ⟨ SOS ⟩ . Given the input start sequence ,it calculate the probabilities of all the words and selects the top three words.

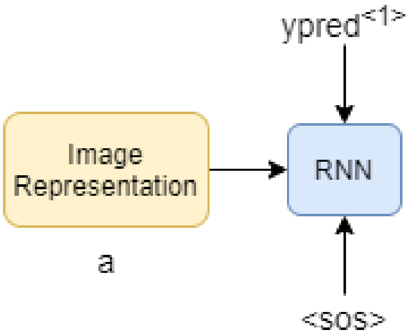


Figure 13.1: Beam Search Step 1

The equation is represented as follows

$$P(ypred^{<1>} \mid x, a) = [The, Little, Girl] \tag{13.1}$$

Now after selecting the first 3 words, it will calculate second from considering all the 3 words. It will calculate the probabilities given the new word as input for all 3 words.

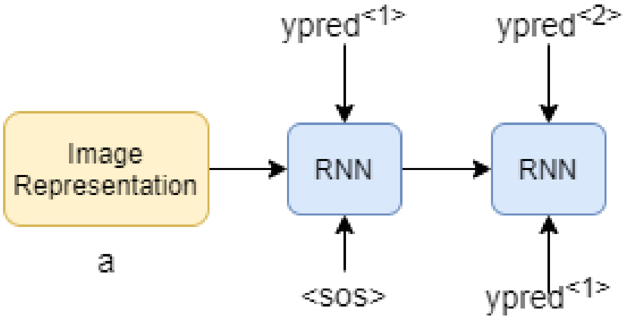


Figure 13.2: Beam Search Step 2

This is mathematically represented in from of equations as follows

$$P(ypred^{<2>} \mid "The", x, a) \tag{13.2}$$

$$P(ypred^{<2>} \mid "Little", x, a) \tag{13.3}$$

$$P(ypred^{<2>} \mid "Girl", x, a) \tag{13.4}$$

Here we eliminate the word ”Girl” from first word list. As beam width is 3,each step instantiates three copies of network to evaluate these partial sentence fragments and outputs.AT given moment in time there will be n2 branches.This process will iterate till any one of the branch reaches end of statement $\langle$ EOS $\rangle$.The Beam search caption generation process can be represented in tree structure as follows:

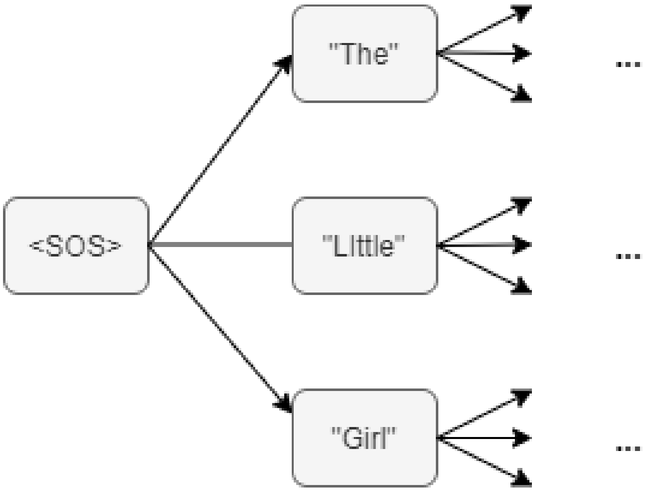


Figure 13.3: Beam Search Tree Representation

We measured the performance of Beam Search with different values of Beam Width on 500 testing examples.

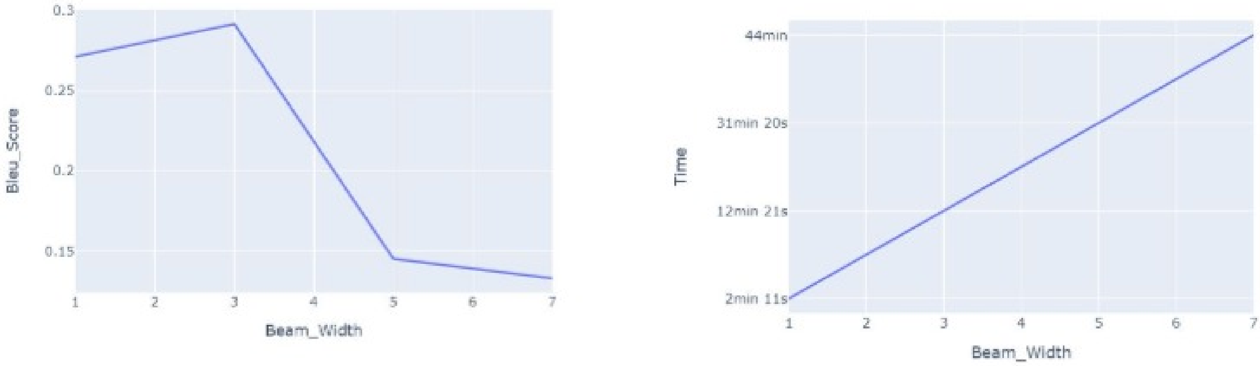
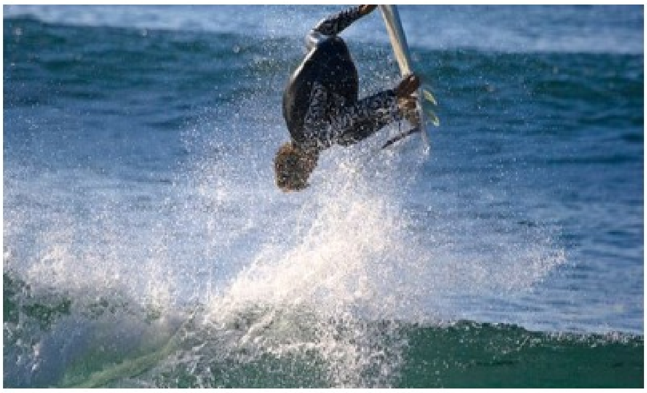


Figure 13.4: Beam Search Tree Representation



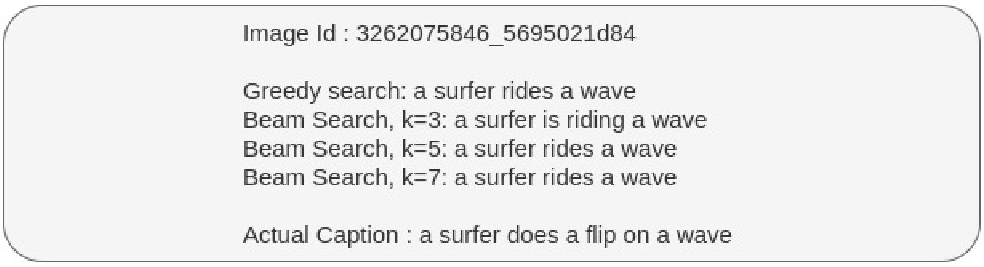
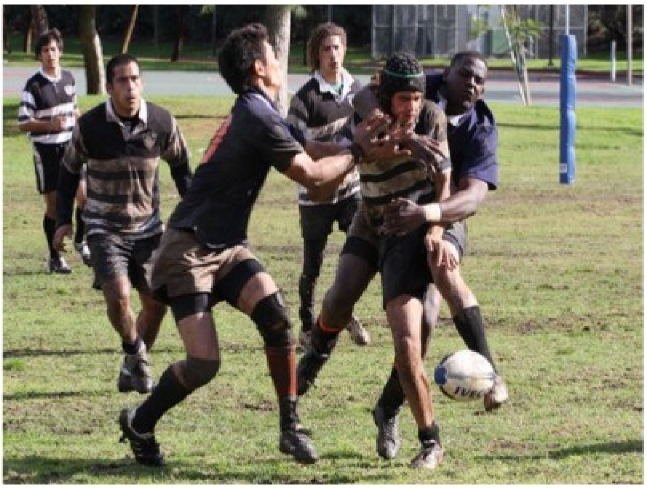


Image Id : 3262075846_5695021d84

Greedy search: a surfer rides a wave
Beam Search, k=3: a surfer is riding a wave
Beam Search, k=5: a surfer rides a wave
Beam Search, k=7: a surfer rides a wave

Actual Caption : a surfer does a flip on a wave



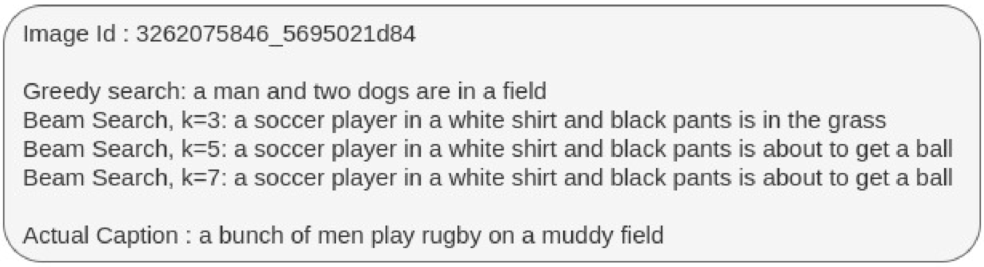


Image Id : 3262075846_5695021d84

Greedy search: a man and two dogs are in a field
Beam Search, k=3: a soccer player in a white shirt and black pants is in the grass
Beam Search, k=5: a soccer player in a white shirt and black pants is about to get a ball
Beam Search, k=7: a soccer player in a white shirt and black pants is about to get a ball

Actual Caption : a bunch of men play rugby on a muddy field

Figure 13.5: Implementation Results of Beam Search

Chapter 14

Evaluation

Now we have to provide a single numerical score to the caption generated which tells us how "good" our translation is when compared to the reference values.

14.1 Bilingual Evaluation Understudy - BLEU Score

BLEU is the standard machine translation (MT) evaluation metric which compares the machine generated translation to the set of good quality reference translations and counts the number of matches in a weighted fashion .BLEU score ranges between 0 and 1. The more the number of matches, the closer the description is to the reference translations, the more BLEU score tends to 1. BLEU score is calculated on n-gram model.

Consider BLEU score on bigrams. The BLEU Score is calculated as follows:

Reference_1 : the girl is climbing into the house
Reference_2 : there is girl near the wooden house
Machine Generated Description : the girl the girl climbing into the house

Bigrams in Translation	Count in Translation	Count_Clip (Count in set of References)
the girl	2	1
girl the	1	0
girl climbing	1	1
the girl	1	1
girl the	1	1
girl climbing	1	1

$$P_n \;=\; \frac{\sum_{ngram}\; Count\_Clip(n - gram)}{\sum_{ngram}\; Count(n - gram)} \tag{14.1}$$

$$P_n \;= \frac{5}{9} \;=\; 0.55 \tag{14.2}$$

Brevity Penalty

To get around the problem of shorter translations, BLEU introduces Brevity Penalty. It penalizes the translation with a length shorter than the length any of our reference translations.

$$Bleu\ Score\ =\ (B_p)\ *\ exp(\frac{1}{n}\ *\sum 1, nP_n) \tag{14.3}$$

where B_p is the Brevity Penalty which is calculated as,

$$\text{f(n)} = \begin{cases} 1 & \text{if } mt_output_length > ref_output_length \\ \exp(1 - \frac{ref_output_length}{mt_output_length}) & otherwise \end{cases}$$

Smoothing Function

If there is no ngrams overlap for any order of n-grams, BLEU returns the value 0. This is because the precision for the order of n-grams without overlap is 0, and the geometric mean in the final BLEU score computation multiplies the 0 with the precision of other n-grams. This results in 0. To avoid this harsh behaviour when no ngram overlaps are found a smoothing function can be used.

14.2 Metric for Evaluation of Translation with Explicit Ordering -METEOR

METEOR matrix uses harmonic mean, unigram precision and recall.It is used as it produces good correlation human generated caption and machine generated caption at segment level.The algorithm takes input the list of hypothesis that is the machine generated captions and reference captions. It first creates a unigram mapping between the given inputs y the following the constraint that every unigram in hypothesis must map to zero or one unigram in references.Once the mapping is completed it calculates the precision and recall values and combine them using harmonic mean to get weighted F score. Here recall is weighted 9 times more than precision.To take into account the word order in hypothesis, we introduce penalty where c is matching chunks and m is number of matches. Using the penalty we get the final Meteor score

14.3 Recall-Oriented Understudy for Gisting Evaluation - ROGUE

ROGUE is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations. To provide quantitative metrics it uses calculates the following :

Recall

Recall tells us how much of the reference summary is captured by the machine generated translation

$$Recall = \frac{numberofoverlappinginreferenceandhypothesis}{totalnumberofwordsinreference} \tag{14.4}$$

A machine generated caption can be extremely long, capturing all words in the reference. But, many of the words in the hypothesis may be useless, making the summary unnecessarily verbose.

Precision

$$Precision = \frac{numberofoverlappinginreferenceandhypothesis}{totalnumberofwordsinhypothesis} \tag{14.5}$$

F-Score

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{14.6}$$

Based on the granularity of the text, ROUGE can be classified as :

- ROUGE-N — measures unigram, bigram, trigram and higher order n-gram overlap
- ROUGE-L — measures longest matching sequence of words using LCS.
- ROUGE-S — Is any pair of words in a sentence in order, allowing for arbitrary gaps. This can also be called skip-gram concurrence.

14.4 Results of Our Model

The BLEU Score evaluation results on our implemented InceptionV3 - LSTM architectures with different relationships between CNN-RNN are :

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	F ROGUE	P ROGUE	R ROGUE
Init-Inject	0.4470	0.3277	0.2836	0.2492	0.28468	0.1538	0.666	0.1666
Pre-Inject	0.41755	0.31022	0.27217	0.2399	0.30323	0.2020	0.2169	0.2029
Par-Inject	0.4361	0.3179	0.2717	0.2443	0.31089	0.2050	0.2139	0.2171
Merge	0.5603	0.3732	0.3101	0.2680	0.13541	0.1408	0.1769	0.2171

Table 14.1: Evaluation of Implemented Models

14.4.1 Init-Inject

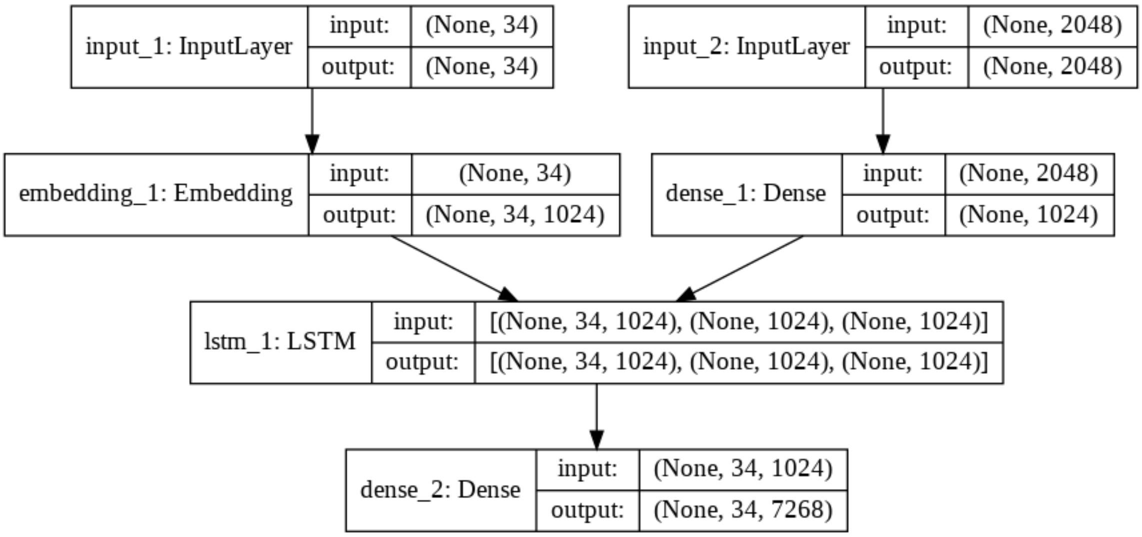


Figure 14.1: Init-Inject : Training Model

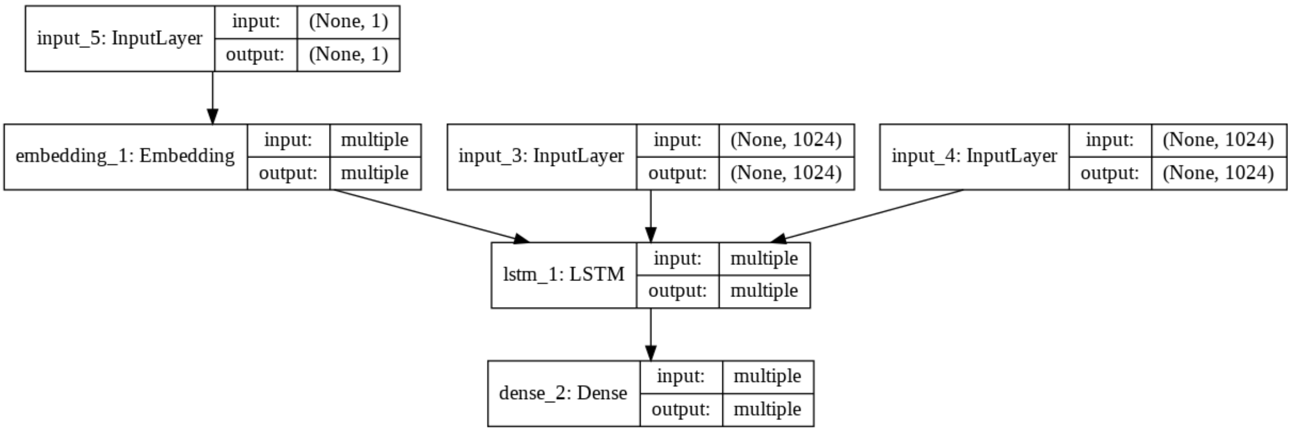


Figure 14.2: Init-Inject : Generation Model

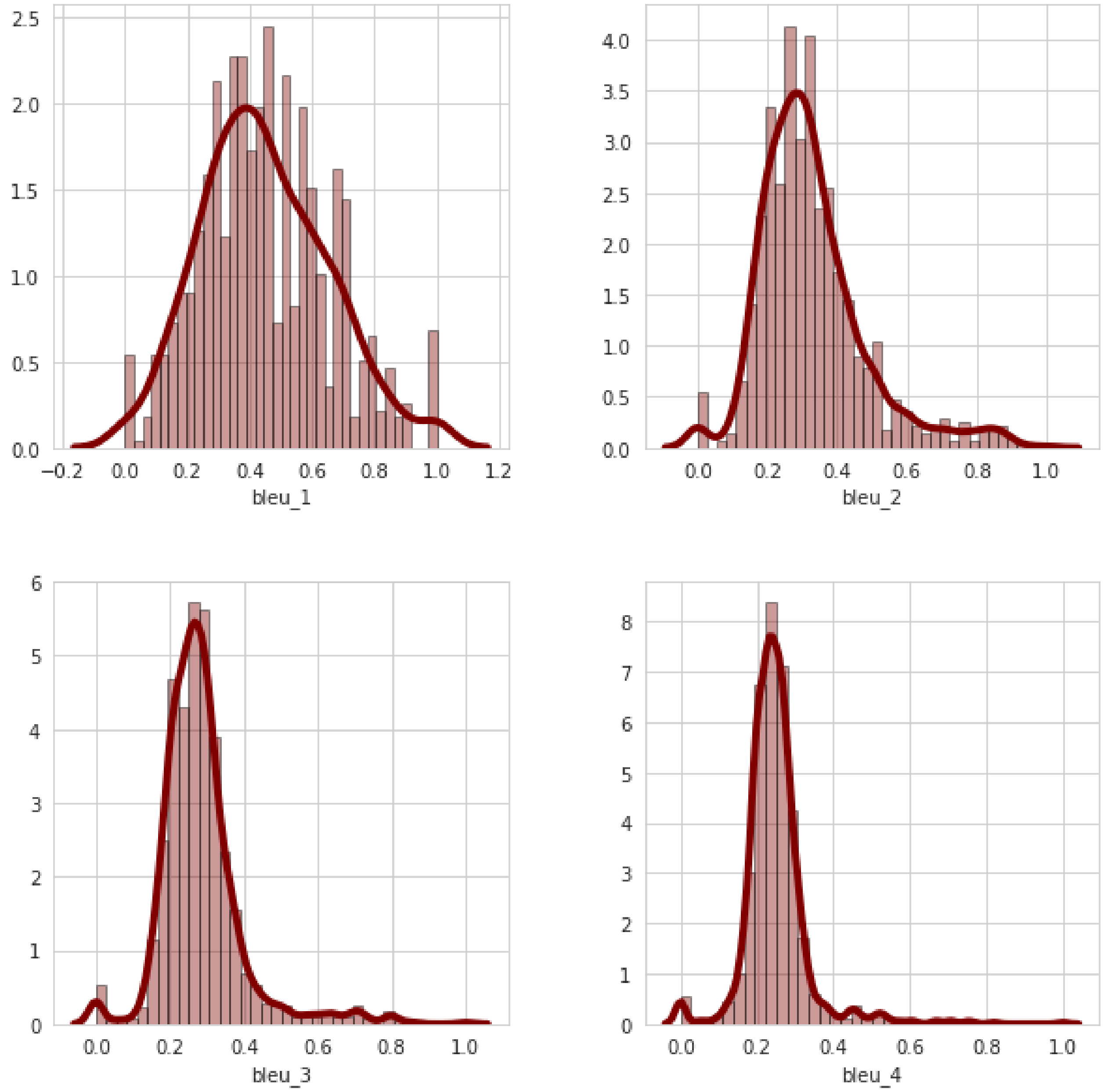
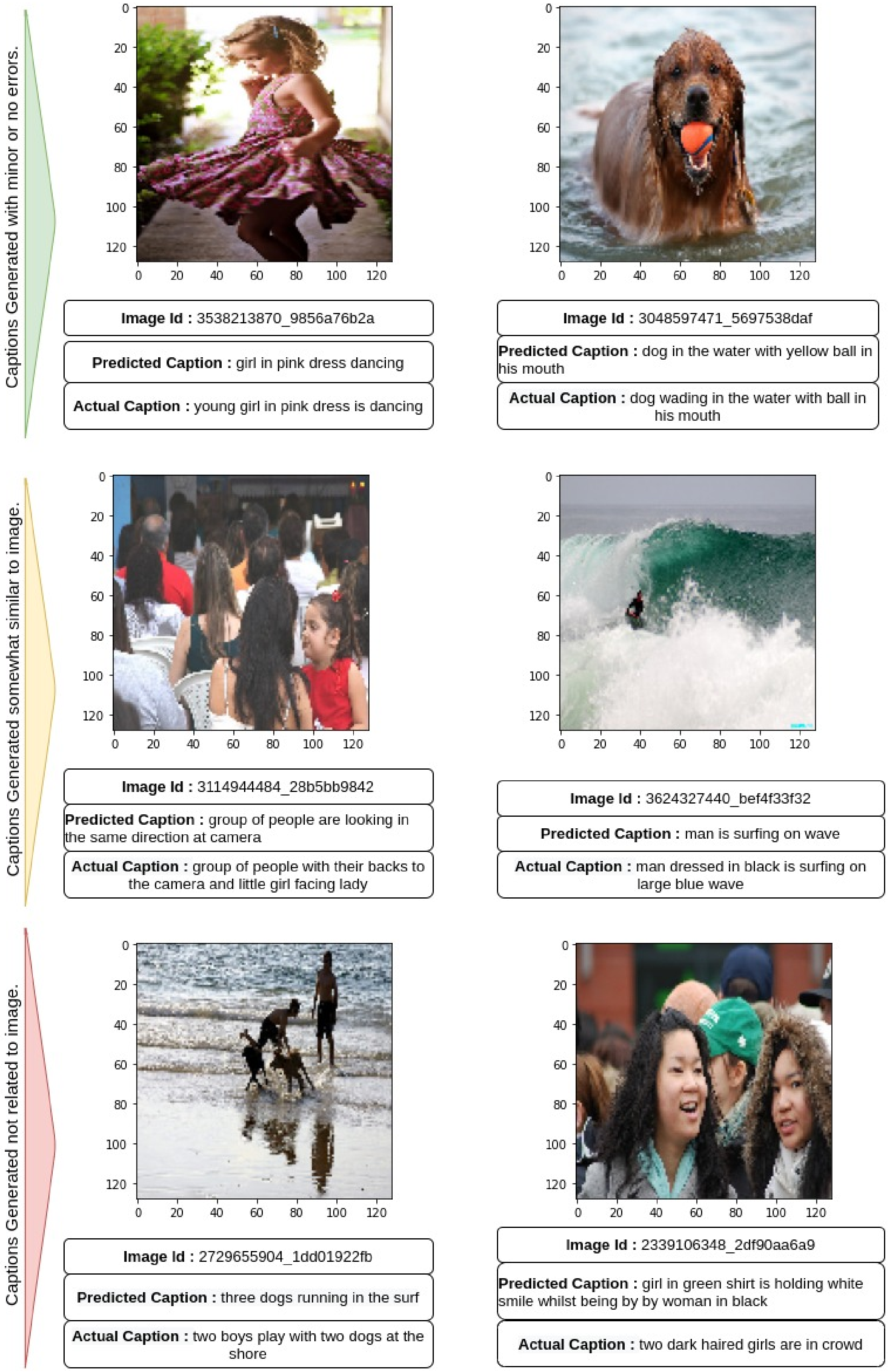


Figure 14.3: Init-Inject : Distribution of Bleu Score



Captions Generated with minor or no errors.

Image Id : 3538213870_9856a76b2a

Predicted Caption : girl in pink dress dancing

Actual Caption : young girl in pink dress is dancing

Image Id : 3048597471_5697538daf

Predicted Caption : dog in the water with yellow ball in his mouth

Actual Caption : dog wading in the water with ball in his mouth

Captions Generated somewhat similar to image.

Image Id : 3114944484_28b5bb9842

Predicted Caption : group of people are looking in the same direction at camera

Actual Caption : group of people with their backs to the camera and little girl facing lady

Image Id : 3624327440_bef4f33f32

Predicted Caption : man is surfing on wave

Actual Caption : man dressed in black is surfing on large blue wave

Captions Generated not related to image.

Image Id : 2729655904_1dd01922fb

Predicted Caption : three dogs running in the surf

Actual Caption : two boys play with two dogs at the shore

Image Id : 2339106348_2df90aa6a9

Predicted Caption : girl in green shirt is holding white smile whilst being by by woman in black

Actual Caption : two dark haired girls are in crowd

Figure 14.4: Init-Inject : Captions Generated

14.4.2 Pre-Inject

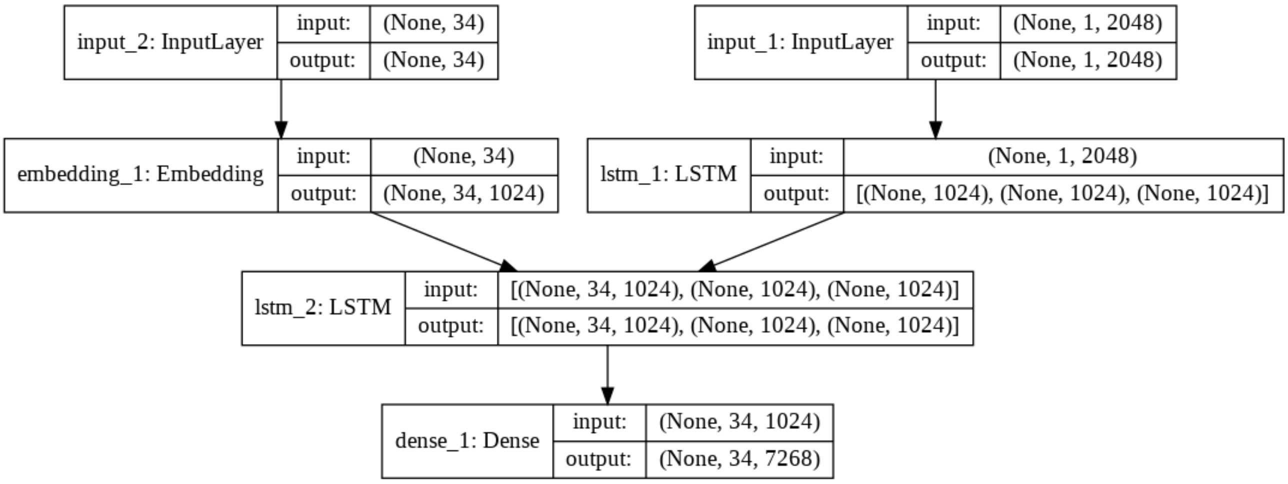


Figure 14.5: Pre-Inject : Training Model

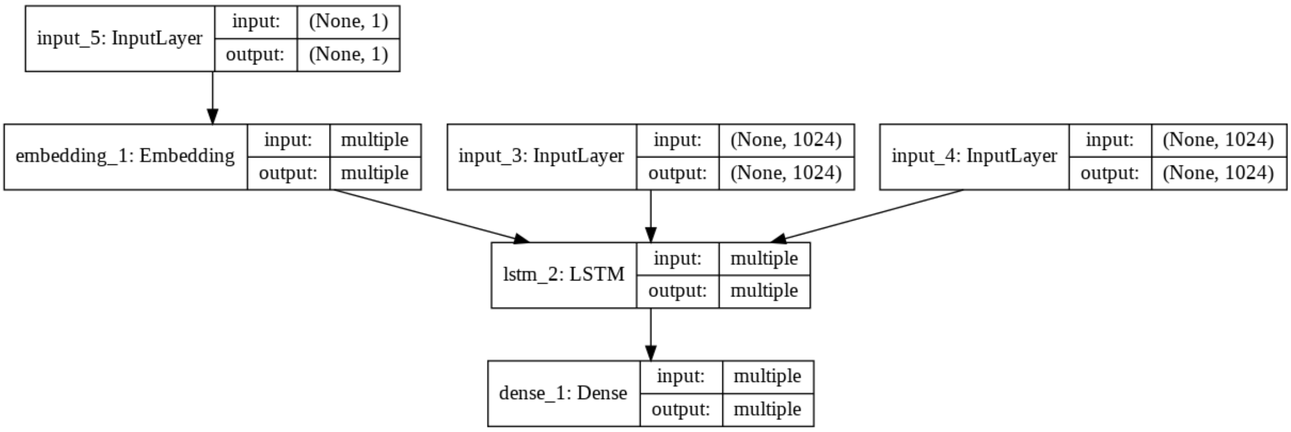


Figure 14.6: Pre-Inject : Generation Model

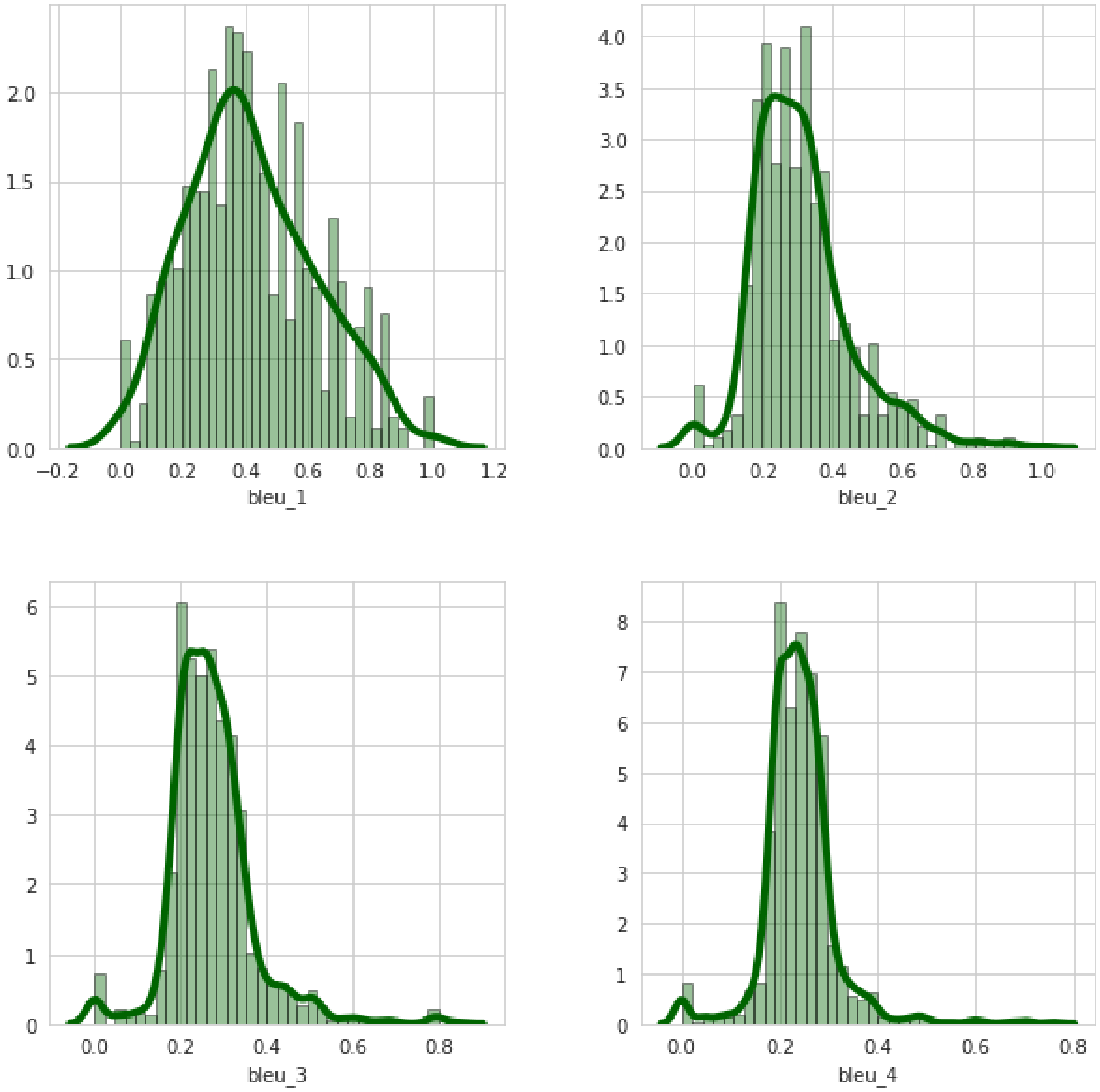


Figure 14.7: Pre-Inject : Distribution of Bleu Score

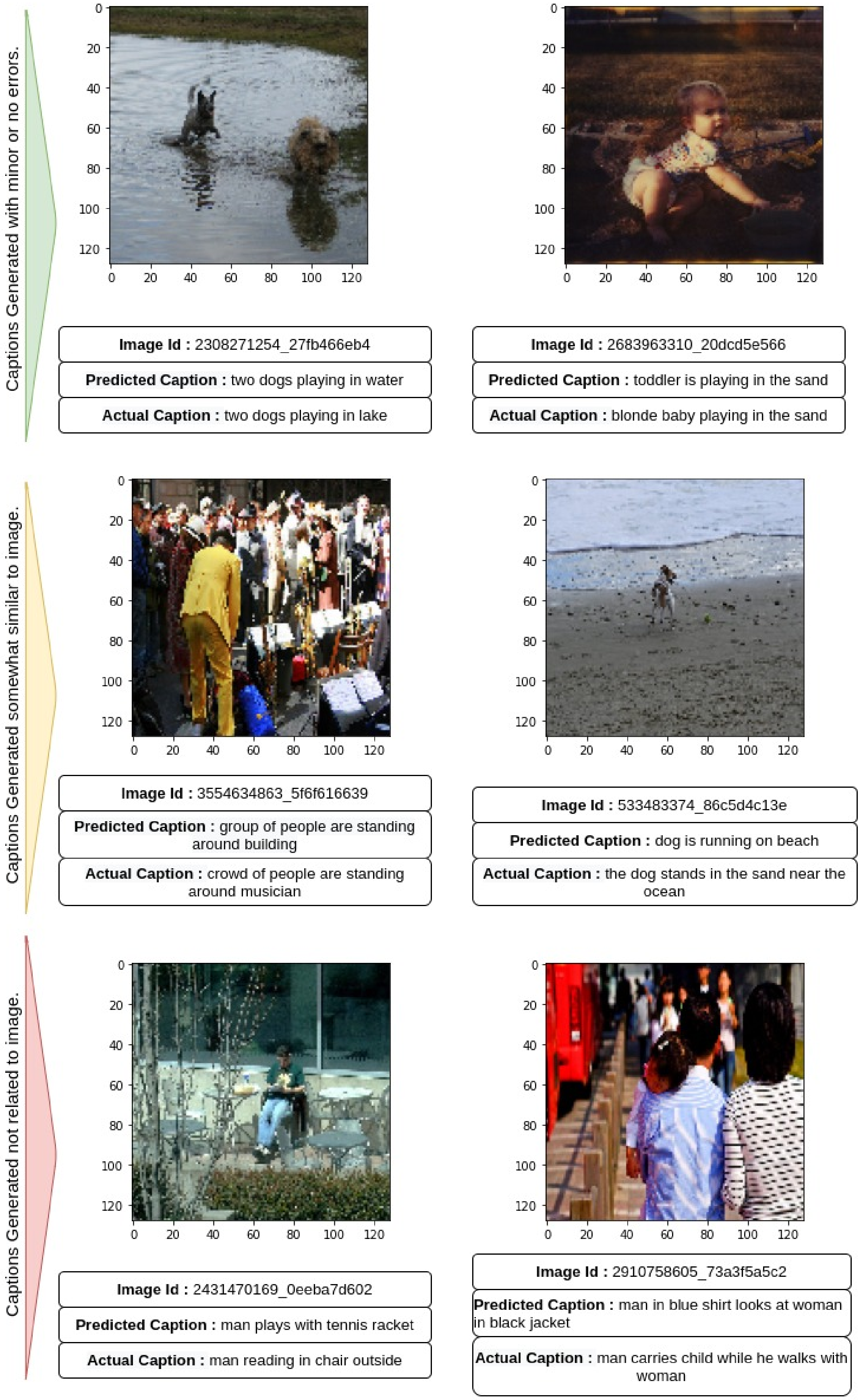

Captions Generated with minor or no errors.
Captions Generated somewhat similar to image.
Captions Generated not related to image.
0
20
40
60
80
100
120

Image Id : 2308271254_27fb466eb4

Predicted Caption : two dogs playing in water

Actual Caption : two dogs playing in lake

Image Id : 2683963310_20dcd5e566

Predicted Caption : toddler is playing in the sand

Actual Caption : blonde baby playing in the sand

Image Id : 3554634863_5f6f616639

Predicted Caption : group of people are standing around building

Actual Caption : crowd of people are standing around musician

Image Id : 533483374_86c5d4c13e

Predicted Caption : dog is running on beach

Actual Caption : the dog stands in the sand near the ocean

Image Id : 2431470169_0eeba7d602

Predicted Caption : man plays with tennis racket

Actual Caption : man reading in chair outside

Image Id : 2910758605_73a3f5a5c2

Predicted Caption : man in blue shirt looks at woman in black jacket

Actual Caption : man carries child while he walks with woman

Figure 14.8: Pre-Inject : Captions Generated

14.4.3 Par-Inject

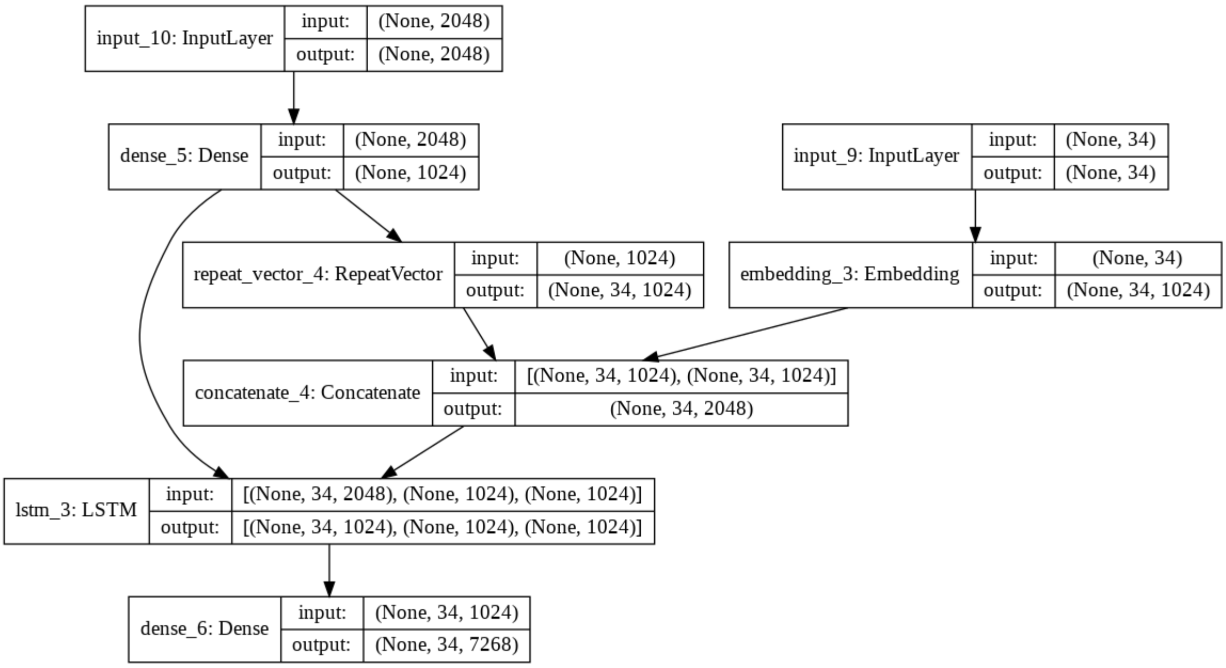

input_10: InputLayer | input: (None, 2048) | output: (None, 2048)
dense_5: Dense | input: (None, 2048) | output: (None, 1024)
input_9: InputLayer | input: (None, 34) | output: (None, 34)
repeat_vector_4: RepeatVector | input: (None, 1024) | output: (None, 34, 1024)
embedding_3: Embedding | input: (None, 34) | output: (None, 34, 1024)
concatenate_4: Concatenate | input: [(None, 34, 1024), (None, 34, 1024)] | output: (None, 34, 2048)
lstm_3: LSTM | input: [(None, 34, 2048), (None, 1024), (None, 1024)] | output: [(None, 34, 1024), (None, 1024), (None, 1024)]
dense_6: Dense | input: (None, 34, 1024) | output: (None, 34, 7268)

Figure 14.9: Par-Inject : Training Model

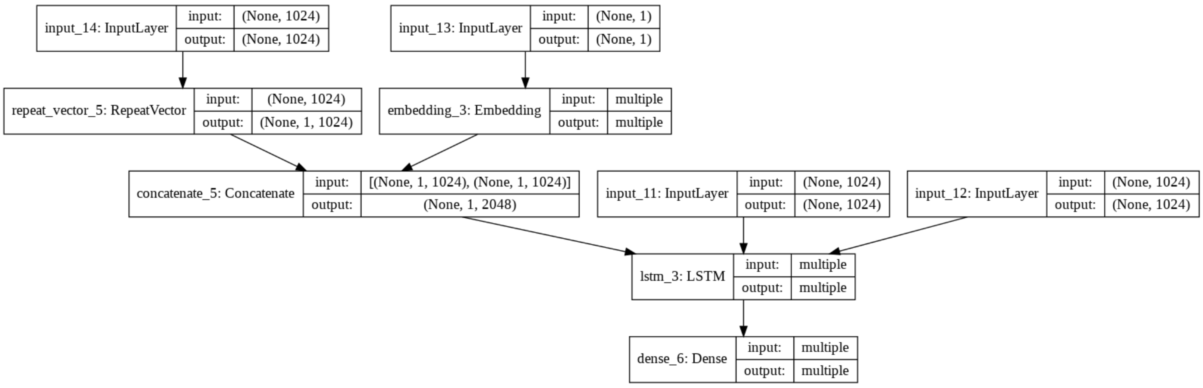

input_14: InputLayer | input: (None, 1024) | output: (None, 1024)
input_13: InputLayer | input: (None, 1) | output: (None, 1)
repeat_vector_5: RepeatVector | input: (None, 1024) | output: (None, 1, 1024)
embedding_3: Embedding | input: multiple | output: multiple
concatenate_5: Concatenate | input: [(None, 1, 1024), (None, 1, 1024)] | output: (None, 1, 2048)
input_11: InputLayer | input: (None, 1024) | output: (None, 1024)
input_12: InputLayer | input: (None, 1024) | output: (None, 1024)
lstm_3: LSTM | input: multiple | output: multiple
dense_6: Dense | input: multiple | output: multiple

Figure 14.10: Par-Inject : Generation Model

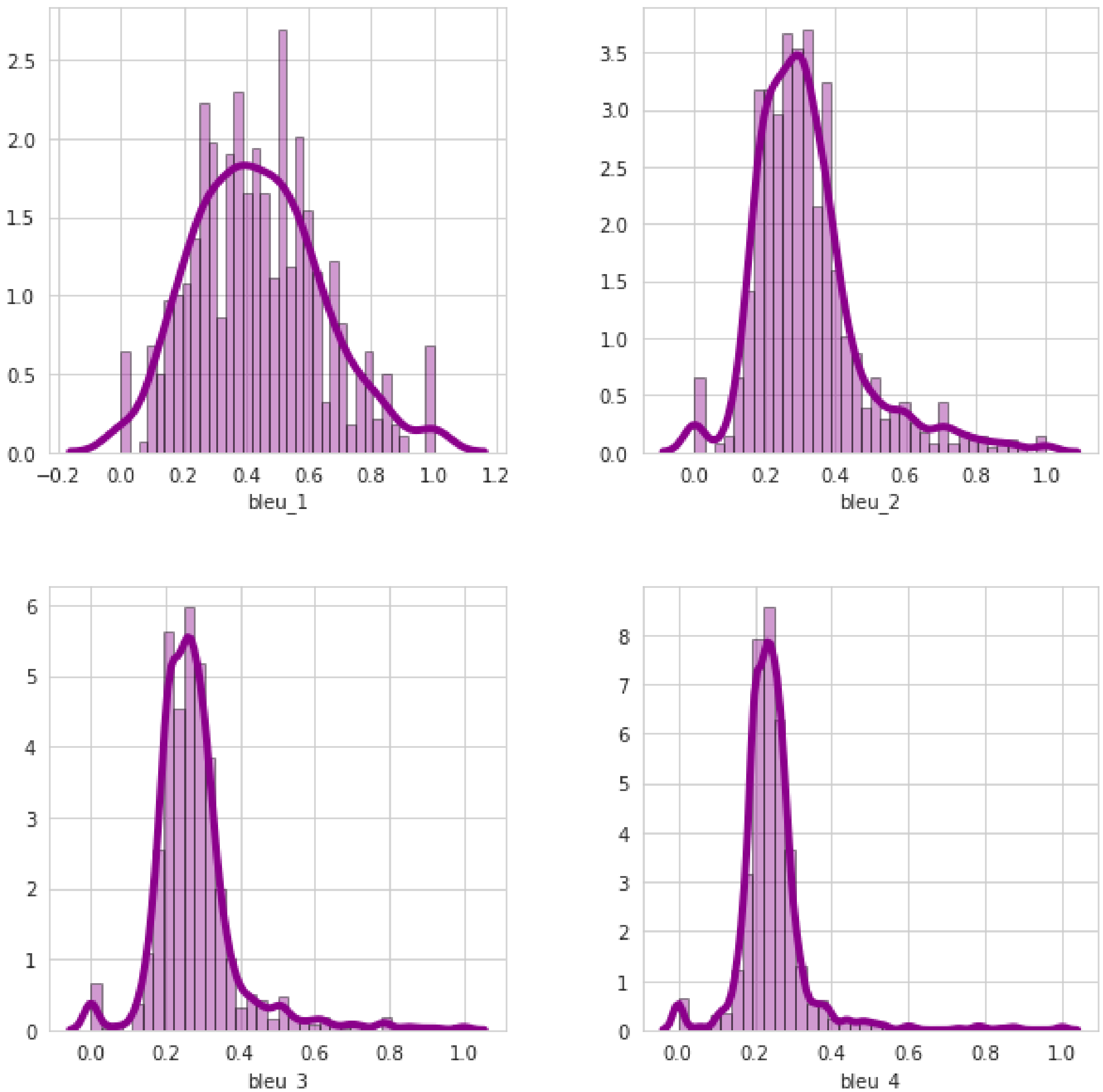
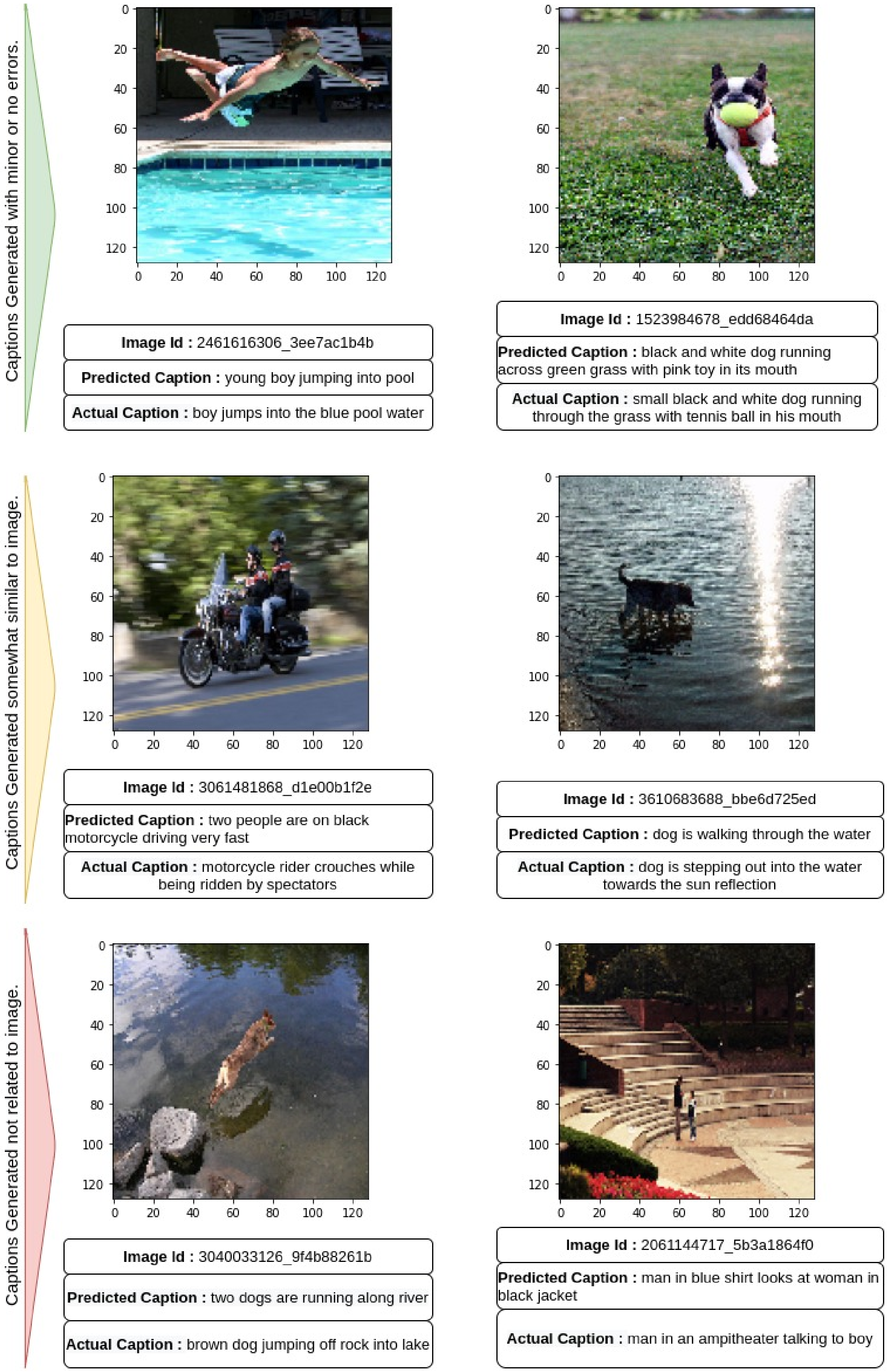


Figure 14.11: Par-Inject : Distribution of Bleu Score



Captions Generated with minor or no errors.

Image Id : 2461616306_3ee7ac1b4b

Predicted Caption : young boy jumping into pool

Actual Caption : boy jumps into the blue pool water

Image Id : 1523984678_edd68464da

Predicted Caption : black and white dog running across green grass with pink toy in its mouth

Actual Caption : small black and white dog running through the grass with tennis ball in his mouth

Captions Generated somewhat similar to image.

Image Id : 3061481868_d1e00b1f2e

Predicted Caption : two people are on black motorcycle driving very fast

Actual Caption : motorcycle rider crouches while being ridden by spectators

Image Id : 3610683688_bbe6d725ed

Predicted Caption : dog is walking through the water

Actual Caption : dog is stepping out into the water towards the sun reflection

Captions Generated not related to image.

Image Id : 3040033126_9f4b88261b

Predicted Caption : two dogs are running along river

Actual Caption : brown dog jumping off rock into lake

Image Id : 2061144717_5b3a1864f0

Predicted Caption : man in blue shirt looks at woman in black jacket

Actual Caption : man in an ampitheater talking to boy

Figure 14.12: Par-Inject : Captions Generated

14.4.4 Merge Model

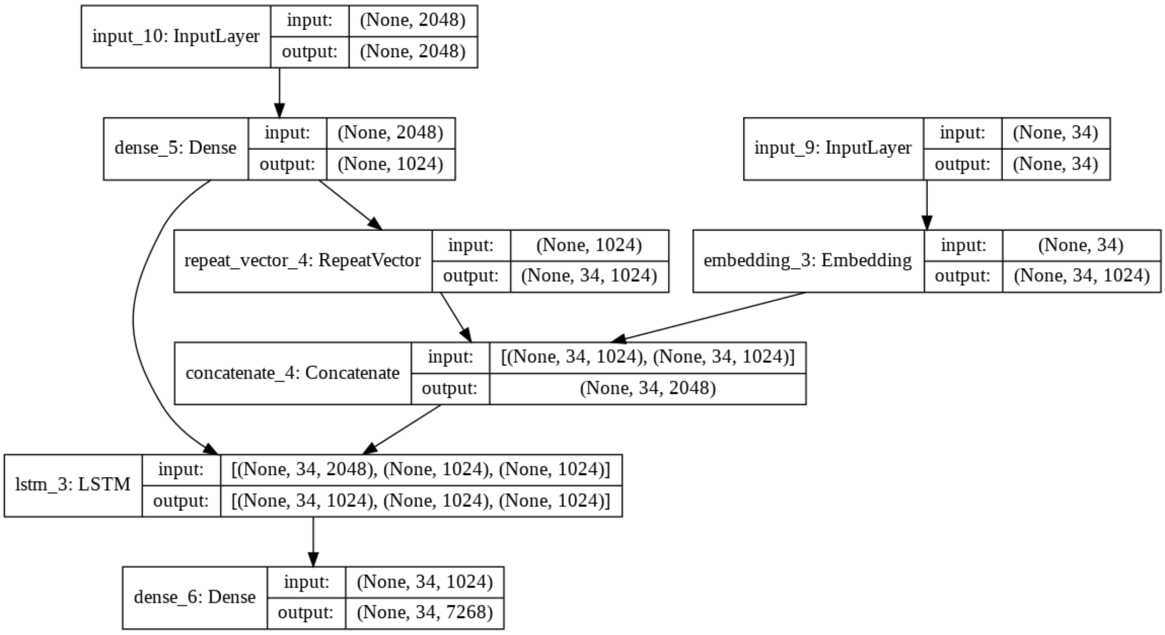


Figure 14.13: Merge : Training and Generation Model

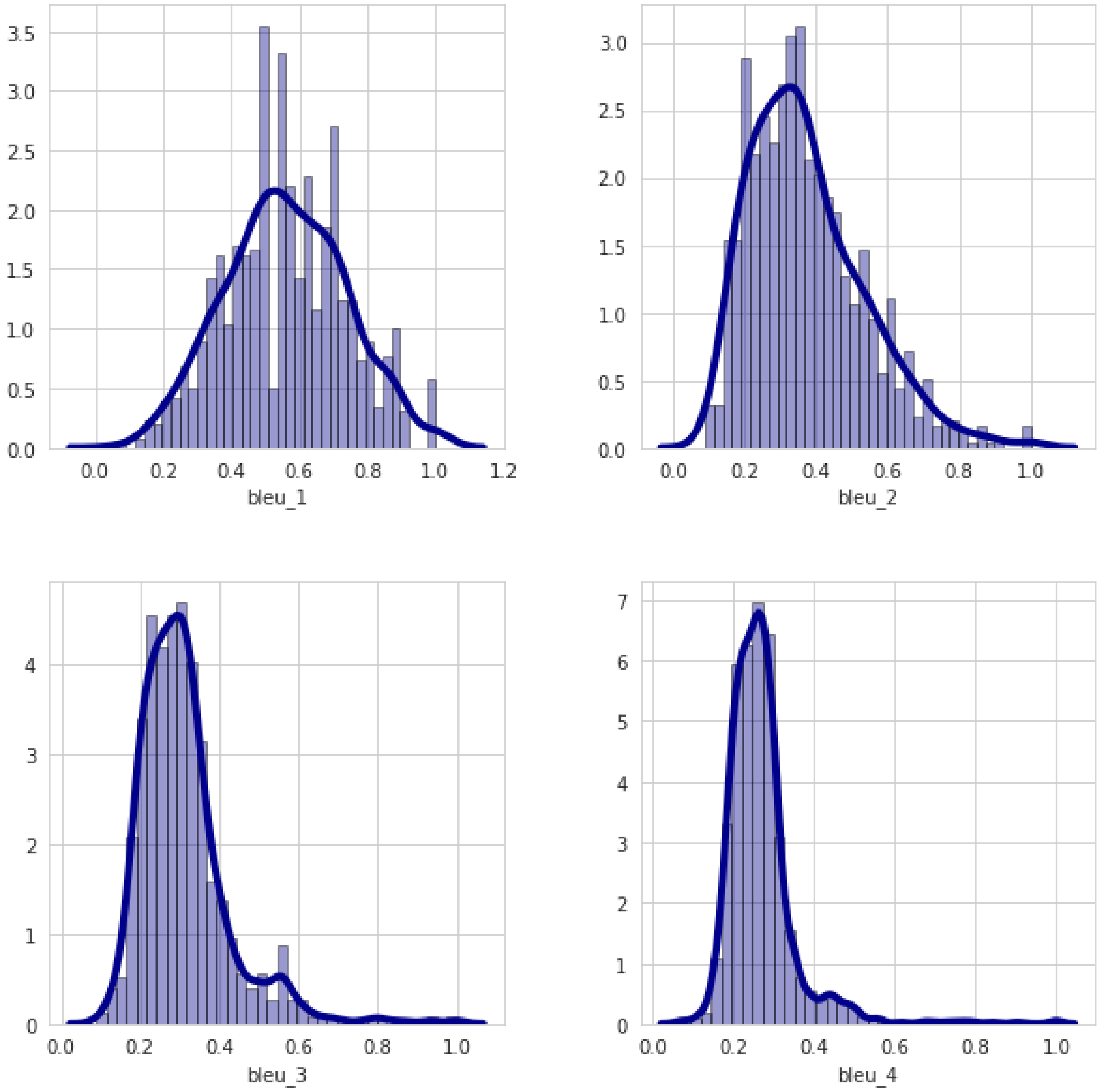
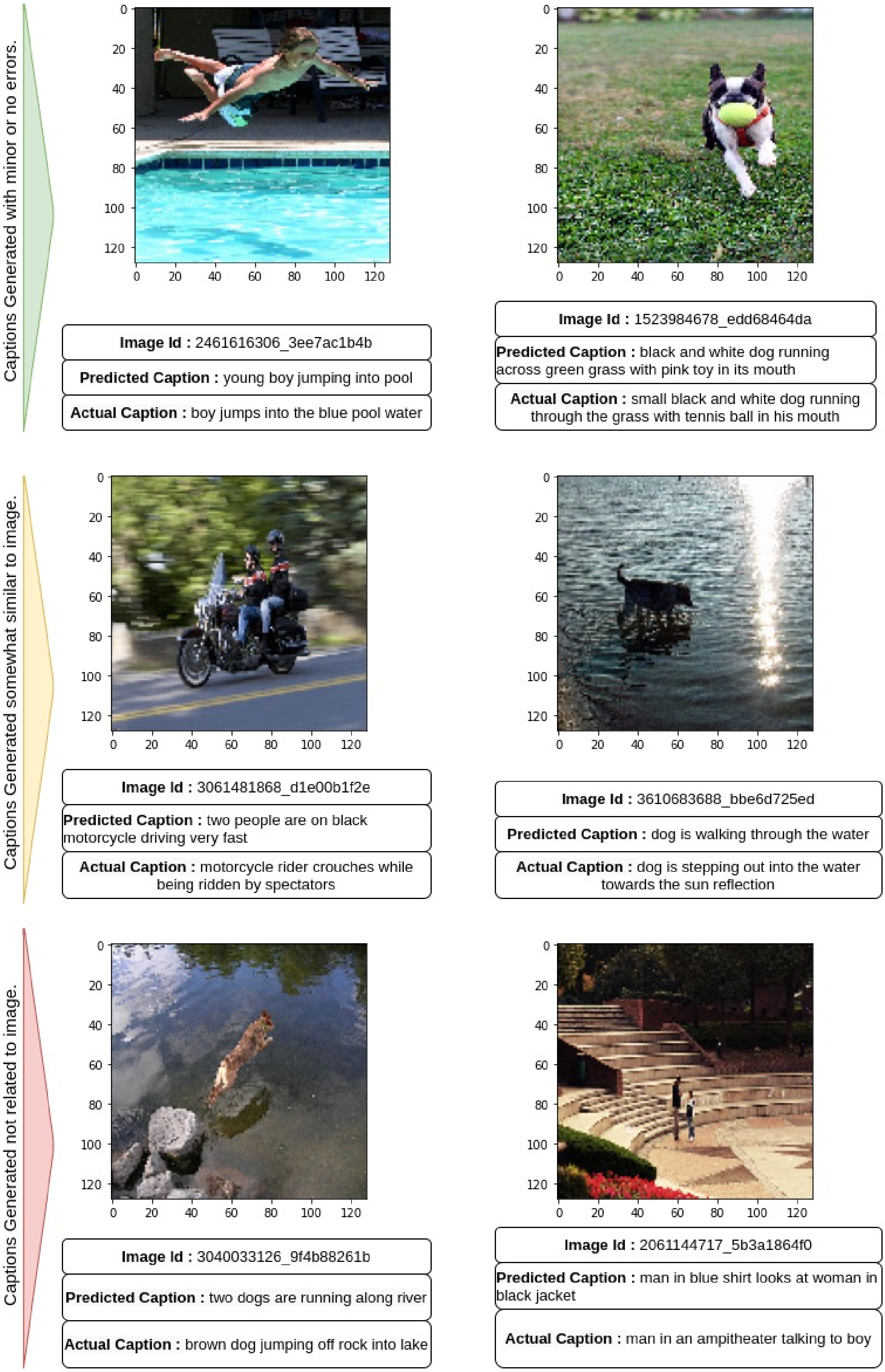


Figure 14.14: Merge : Distribution of Bleu Score



Captions Generated with minor or no errors.

Image Id : 2461616306_3ee7ac1b4b

Predicted Caption : young boy jumping into pool

Actual Caption : boy jumps into the blue pool water

Image Id : 1523984678_edd68464da

Predicted Caption : black and white dog running across green grass with pink toy in its mouth

Actual Caption : small black and white dog running through the grass with tennis ball in his mouth

Captions Generated somewhat similar to image.

Image Id : 3061481868_d1e00b1f2e

Predicted Caption : two people are on black motorcycle driving very fast

Actual Caption : motorcycle rider crouches while being ridden by spectators

Image Id : 3610683688_bbe6d725ed

Predicted Caption : dog is walking through the water

Actual Caption : dog is stepping out into the water towards the sun reflection

Captions Generated not related to image.

Image Id : 3040033126_9f4b88261b

Predicted Caption : two dogs are running along river

Actual Caption : brown dog jumping off rock into lake

Image Id : 2061144717_5b3a1864f0

Predicted Caption : man in blue shirt looks at woman in black jacket

Actual Caption : man in an ampitheater talking to boy

Figure 14.15: Merge : Captions Generated

Chapter 15

Conclusions

Key Takeaways

This Project of Automatic Generation of Descriptions from an image has been a major learning curve from the perspective of Computer Vision as well as Natural Language Processing. A vast amount of work is done in this field. Through this project we gained an in-depth understanding of the approaches used in the literature to solve this problem.
We developed a solid knowledge of deep learning, NLP and CV, combining both theory knowledge with hands-on experience, techniques to develop and improve Deep Networks, Convolution Neural Networks, LSTM, and Sequence to Sequence modeling.This project aimed at learning various approaches to solve the task, compare various approaches for conditioning the language model with image features,heuristic search methods, technologies to implement them rather than maximizing the evaluation results.

- In our observations, the bleu score evaluation of merge architecture is better than the inject architecture. However the evaluation metrics differed only by few points. Choosing merge or inject architectures are not achingly unhealthy to the model's performance.

- The visual and linguistic embedding can be combined at an initial state or can be delayed to the later stage.Thus, RNN can be viewed as a generator as well as an encoder in caption generation model.

- To train the model the data can be structured to generate partial captions upto time$\langle$t$\rangle$ to train for predicting the word at time$\langle$t+1$\rangle$. Modelling the data in this format is however memory intensive.

- Beam search is an important search algorithm that explore the tree for the promising nodes. As the beam width increases, along with time taken to search for an optimal translation, the requirements for memory also scale up. Also, we observed that as the beam width increases the optimal hypothesis found by beam search degrades. On our training, the use of narrower beam width provided better translations.

Chapter 16

Future Scope

We would like to continue and extend our work on following points.

- Tailoring this model to focus on the user-specific needs and deliver the description would be a challenging task. We would love to extend this work on an application based on Context Based Image Retrieval.

- Attention Mechanism
  Various attention mechanisms have been added to the deep learning models for improving generation of the captions. Using Attention Mechanisms the model can focus on the salient elements of the image while generating the corresponding description of the region. The successes of employing Attention Mechanism has encouraged us to continue work on this problem by conducting Experiments on Attention Mechanisms in Caption Generation.

- Learning Other Approaches.
  Developing a model with the recent advances like Reinforcement Learning and Generative Adversarial Networks(GANs) is challenging and intriguing. Reinforcement learning methods have been proposed to maximize the language quality of the Captions generated. Similarly, GANs have been introduced with an aim to improve the naturalness and diversity of generated captions.

Bibliography

[1] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., and Chua, T.-S. (2017a). SCA-CNN:Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6298–6306, Honolulu, USA. IEEE.

[2] Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., and Forsyth, D. (2010). Every Picture Tells a Story: Generating Sentences from Images. In Proceedings of the 11th European Conference on Computer Vision (ECCV), volume 6314 LNCS, pages 15–29, Heraklion, Greece. Elsevier.

[3] Gupta, A., Verma, Y., and Jawahar, C. V. (2012). Choosing linguistics over vision to describe images. In Proceedings of the Twenty-Sixth (AAAI) Conference on Artificial Intelligence, pages 606–612, Toronto, Ontario, Canada. AAAI Press.

[4] Jia, X., Gavves, E., Fernando, B., and Tuytelaars, T. (2015). Guiding the Long-Short Term Memory Model for Image Caption Generation. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 2407–2415, Santiago, Chile. IEEE.

[5] Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014a). Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. In Advances in Neural Information Processing Systems Deep Learning Workshop.

[6] Kiros, R., Zemel, R., and Salakhutdinov, R. (2014b). Multimodal Neural Language Models. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), volume 32, pages 595–603, Beijing, China. Proceedings of Machine Learning Research.

[7] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2011). Baby talk: Understanding and generating simple image descriptions. In 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), volume 18, pages 1601–1608, Colorado Springs, USA. IEEE.

[8] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014), volume 8693 LNCS, pages 740–755. Springer.

[9] Ma, S. and Han, Y. (2016). Describing images by feeding LSTM with structural words. In 2016 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6. IEEE.

[10] Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2014). Explain Images with Multimodal Recurrent Neural Networks. In NIPS 2014 Deep Learning Workshop.

[11] Mao, J. and Yuille, A. (2015). Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). 2015 International Conference on Learning Representations (ICLR 2015), 1090(2014):1–17.

[12] Oliva, A. and Torralba, A. (2006). Building the gist of a scene: the role of global image features in recognition. In Progress of Brain Research, volume 155, pages 23–36. Elsevier B.V

[13] Ordonez, V., Kulkarni, G., and Berg, T. (2011). Im2text: Describing Images Using 1 Million Captioned Photographs. Advances in Neural Information Processing Systems (NIPS), pages 1143–1151.

[14] Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. (2010). Collecting Image Annotations Using Amazon's Mechanical Turk. In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk,, pages 139–147, Los Angeles, USA.

[bm] harma, P., Ding, N., Goodman, S., and 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, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics

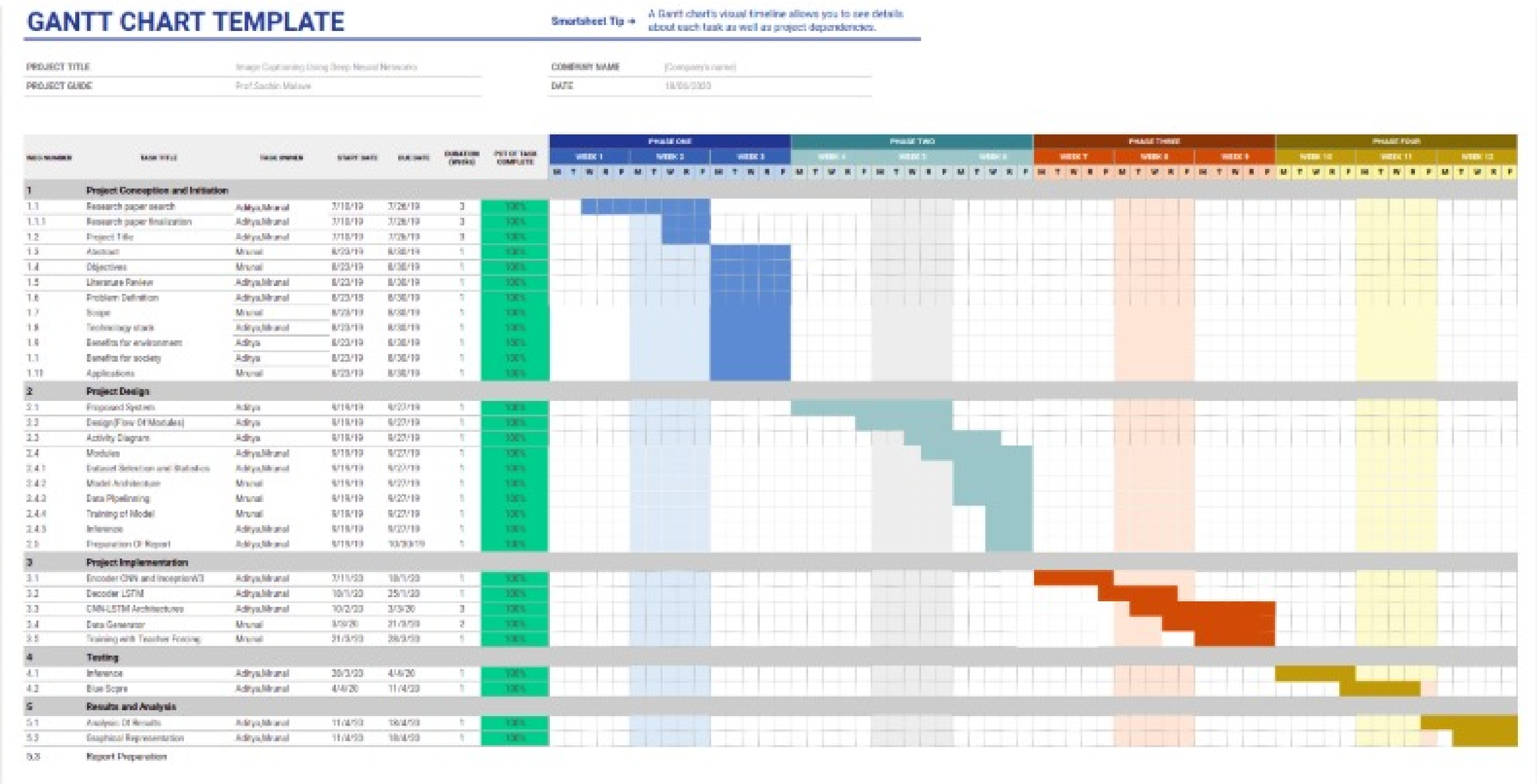
[15] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In Advances in Neural Information Processing Systems 27 (NIPS 2014), pages 3104–3112.

[16] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 07-12-June, pages 3156–3164, Boston, USA. IEEE.

[17] Yang, Y., Teo, C., Daumé III, H., and Aloimonos, Y. (2011). Corpus-Guided Sentence Generation of Natural Images. In The 2011 Conference on Empirical Methods in Natural Language Processing, pages 444–454.

[18] Young, M. H. P., Lai, A., and 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.

[19] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In 32nd International Conference on Machine Learning (ICML'15), volume 37, pages 2048–2057, Lille, France.



GANTT CHART TEMPLATE

Smartsheet Tip → A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE	Image Captioning Using Deep Neural Networks	COMPANY NAME	[Company's name]
PROJECT GUIDE	Prof Sachin Malave	DATE	18/06/2020

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (WEEKS)	PCT OF TASK COMPLETE
1	**Project Conception and Initiation**					
1.1	Research paper search	Aditya,Mrunal	7/18/19	7/28/19	3	100%
1.1.1	Research paper finalization	Aditya,Mrunal	7/18/19	7/28/19	3	100%
1.2	Project Title	Aditya,Mrunal	7/18/19	7/28/19	3	100%
1.3	Abstract	Mrunal	8/23/19	8/30/19	1	100%
1.4	Objectives	Mrunal	8/23/19	8/30/19	1	100%
1.5	Literature Review	Aditya,Mrunal	8/23/19	8/30/19	1	100%
1.6	Problem Definition	Aditya,Mrunal	8/23/19	8/30/19	1	100%
1.7	Scope	Mrunal	8/23/19	8/30/19	1	100%
1.8	Technology stack	Aditya,Mrunal	8/23/19	8/30/19	1	100%
1.9	Benefits for environment	Aditya	8/23/19	8/30/19	1	100%
1.1	Benefits for society	Aditya	8/23/19	8/30/19	1	100%
1.11	Applications	Mrunal	8/23/19	8/30/19	1	100%
2	**Project Design**					
2.1	Proposed System	Aditya	9/19/19	9/27/19	1	100%
2.2	Design(Flow Of Modules)	Aditya	9/19/19	9/27/19	1	100%
2.3	Activity Diagram	Aditya	9/19/19	9/27/19	1	100%
2.4	Modules	Aditya,Mrunal	9/19/19	9/27/19	1	100%
2.4.1	Dataset Selection and Statistics	Aditya,Mrunal	9/19/19	9/27/19	1	100%
2.4.2	Model Architecture	Mrunal	9/19/19	9/27/19	1	100%
2.4.3	Data Pipelining	Mrunal	9/19/19	9/27/19	1	100%
2.4.4	Training of Model	Mrunal	9/19/19	9/27/19	1	100%
2.4.5	Inference	Aditya,Mrunal	9/19/19	9/27/19	1	100%
2.5	Preparation Of Report	Aditya,Mrunal	9/19/19	10/30/19	1	100%
3	**Project Implementation**					
3.1	Encoder CNN and InceptionV3	Aditya,Mrunal	7/1/20	10/1/20	1	100%
3.2	Decoder LSTM	Aditya,Mrunal	10/1/20	25/1/20	1	100%
3.3	CNN-LSTM Architectures	Aditya,Mrunal	10/2/20	3/3/20	3	100%
3.4	Data Generator	Mrunal	3/3/20	21/3/20	2	100%
3.5	Training with Teacher Forcing	Mrunal	21/3/20	28/3/20	1	100%
4	**Testing**					
4.1	Inference	Aditya,Mrunal	28/3/20	4/4/20	1	100%
4.2	Blue Score	Aditya,Mrunal	4/4/20	11/4/20	1	100%
5	**Results and Analysis**					
5.1	Analysis Of Results	Aditya,Mrunal	11/4/20	18/4/20	1	100%
5.2	Graphical Representation	Aditya,Mrunal	11/4/20	18/4/20	1	100%
5.3	Report Preparation					

Figure 16.1: Phase 1 : Gantt Chart

Acknowledgement

We have great pleasure in presenting the report on **Image Captioning Using Deep Neural Networks** We take this opportunity to express our sincere thanks towards our guide **Prof. Sachin Malave** Department of Computer Engineering, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Sachin Malave** Head of Department,Computer Engineering, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof.Amol Kalugade** BE project co-ordinator, Department of Computer Engineering, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1:Mrunal S Jadhav
Student ID1: 16102030

Student Name2: Aditya G Joshi
Student ID2: 16102017