UNDERSTAND & DESCRIBE: A DEEP LEARNING APPROACH FOR IMAGE CAPTIONING IN TELUGU

LEELA GUNA KRISHNA KOMPALLI

A thesis submitted in fulfilment of the

requirements for the degree of

MASTER OF SCIENCE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

LIVERPOOL JOHN MOORES UNIVERSITY (LJMU)

JUNE 2022

DEDICATION

This work is dedicated to my late grandmother ***Indira Devi Kompalli*** and my late aunt ***Sarvani Kompalli*** for their undying love and support. They'll always be remembered.

ACKNOWLEDGEMENTS

Gratitude turns what we have into enough, and more. I'd like to take this opportunity to express my gratitude to ***Mr. Bharath Kumar Bolla*** for being a mentor and guiding me in this research. His outstanding knowledge in the field of Natural Language Processing and his problem-solving skills helped me to overcome many challenges. I'd like to extend my gratitude to my sister ***Uma Pranavi Kompalli*** for assisting me in this research. Finally, I'll thank my parents ***Sreenivasa Murthy Kompalli*** and ***Rupa Rani Kompalli,*** my grandfather ***K. Sesha Talpa Saye Kompalli*** for being with me during my research. I conclude by expressing my special thanks to my friend ***Hamsa Vardhan Darapureddi*** for providing me the motivation I needed.

ABSTRACT

Understanding an image and generating a relevant caption to it has been a challenging task in the field of Artificial Intelligence. This task involves various sub-concepts such as Convolutional Neural Networks (CNNs) for image understanding and Recurrent Neural Networks (RNNs) for text understanding. This task has been carried out by various researchers worldwide, but mainly in the English language. Language should not be a barrier for AI. Hence, in this research, our primary focus would be on observing how well the trained model performs in the Telugu language, i.e., the model has to understand the image and provide a relevant meaning/caption in the Telugu language. Telugu is spoken by around 75 million people around the world, which makes it a morphologically rich language. For this research, we would like to try various methodologies such as Encoder-Decoder Method using LSTMs and GRUs, Attention techniques, and test the BLEU scores, respectively.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CHAPTER 1

INTRODUCTION

1.1 Background

When we see something, we perceive it, understand it, and then our mind generates a language relevant to the picture we have seen. Right from the start of Human evolution, image/concept understanding has been an involuntary act. This act is language independent. The human mind processes the image captured by the human eye and generates a relevant description to whatit perceived. This is called "Visual Question Answering" (VAQ). One of the applications of VAQ is "Image Captioning."

The research on Image Captioning started back in 2002 when the proposed idea was to builda classifier and use basic image processing techniques to get the features. The obtained modelis passed to a lexical model to convert the visual features into a text format. There were manybetter proposals and methods, but the usage of CNNs and LSTMs to generate imagedescriptions in 2015 was the key highlight of this subject.

Generating captions by understanding the images is a subject that is being used in a wide range of applications. From a very useful application called the "Eye to the Blind" to even annotating the images, this subject has a wide scope. It is not just about detecting images or objects but about providing an understanding of the detected image. As this subject requires the knowledge of Image Processing along with Natural Language Processing, it becomes a challenge for the researchers to build an efficient model.

The past research work on this subject has been done in the English Language as the primary reason would be that English is a dominant language. Other than English, image captioning has also been done in various morphologically rich languages such as Chinese, Mandarin, Spanish, French, Hindi, and many others. In this research, we implement Image caption generation in the Telugu language.

Telugu, spoken by around 75 million people worldwide, is also a morphologically rich language. It is one of the six classical languages in India. Telugu is ranked 7^{th} in Asia and 14^{th}in the world based on its total number of speakers. A unique feature we can find in the Telugu language is every word ends with a vowel. It is known as the "Italian of the East," said Niccola

da Conti, a 16th-century Venetian traveler. As the field of AI should not be limited to the language, this research can provide a generalized aspect of how the model understands various languages other than English.

In the South Indian region, there are people who cannot understand the English language and rely on Telugu. This research can provide a path to develop an application that helps native speakers know what they see. For instance, a simple traffic sign can confuse a young native speaker. In such situations, the application can explain or depict the sign with ease. This application can be integrated with a text-to-speech mechanism to help visually impaired native speakers. Apart from these, numerous applications can be developed with the help of this research, such as real-time captioning of videos, assisting passengers in Autonomous vehicles, etc.

1.2 Problem Statement

Marathi has a significant number of text resources and is mostly available in handwritten papers, hence it has a poor digital presence. It has 54 characters in total which is pretty huge for a dataset. Since it's spoken by around 75 million people around the world, this makes Telugu a morphologically rich language.

This research primarily focuses on comparative study of various image captioning techniques in Telugu language. We will understand various visual encoding and language modelling techniques that can be applied to this research.

Finally, we will evaluate the model based on various evaluation techniques such as BLEU scores and attain an efficient model for image captioning in Telugu language.

1.3 Aims and Objectives

In this research, our objective is to understand the provided related works proposed by researchers and develop a model to generate descriptions in the Telugu language by understanding the images. There are two types of approaches, top-down and bottom-up. The top-down approach considers the whole image and provides the respective text analysis. The bottom-up, in contrast, offers a different approach by combining words to form a sentence,

which is relevant to the image. The Encoder-Decoder model built using CNNS and RNNs is based on the above methods. A better way to obtain the efficient model is to use the Attention technique, which focuses on the particular sub-image at a time, reducing the time complexity and increasing the model efficiency.

At the end of this research, we aim to provide a better statistical analysis of how the models using the traditional Encoder-Decoder architecture work compared with the models that use the Attention. We also use different pre-trained CNN models such as VGG-16, Resnet50, InceptionV3 and understand which works better for this research.

1.4 Research Questions

This research tries to cover the below questions/assumptions.

- How well an Encoder-Decoder model built with Long Short-Term Memory Networks(LSTMs) performs compared with Gated Recurrent Units (GRUs)?
- Is the Attention mechanism better than the Encoder-Decoder model?
- Understanding which pre-trained CNN model (Inception-V3, VGG-16, Resnet50)performs better.

1.5 Significance of study:

The motivation to pursue this research is because Artificial Intelligence should not be limited to a language. Most of the research specific to Natural Language Processing is done in English. This limits the study to one particular language. So, in this research, we understand how well a developed model can understand an image and depict Telugu language.

Telugu is the fifth frequently spoken language and the eleventh most spoken in the worldby around 75 million people. It is one of the six classical languages in India. Because of its morphological richness, we aim to develop the Image captioning transformer in Telugu. This task can be a generalized approach to multimodal learning for various languages. This can be developed as a standalone application and can be further improved. For instance, a text-to-speech mechanism can be integrated further to develop an application that aids visually

impaired people.

1.6 Scope of the study

This research can be useful to develop various applications for the visually impaired and native speakers. A virtual assistant can be a better way to guide native speakers. Providing recommendations to products just by observing the images are some of its applications.

Day by day, the usage of digital media is increasing. Different types of art, collection of various images related to astronomy, landscapes, actions are involved in social media. With this enormous amount of data, describing what they are is an important aspect. These can further be developed for real-time description generation.

1.7 Structure of study

This section discusses the thesis report's overall structure. In Chapter 1, we introduced the study issue and established the research's purpose, objective, and inspiration. The research paper intends to address that we have identified in the research questions. For other NLP-related tasks, we also examined the importance of the study in order to detail the ramifications, and wind up with the scope of study.

In Chapter 2, we discuss the literature review which is conducted on this thesis. We examine numerous picture captioning strategies as well as the difficulties of image captioning in Telugu and examine the significance of pre-trained word embeddings, which includes mono and multilingual embeddings. The related research papers that cover all of the strategies utilized for picture captioning for Indian local languages that have been done thus far. However, this chapter concludes with those related research papers.

The research methodology is discussed in depth in Chapter 3. The first portion goes through selecting the data, and also pre-processing methods, and more exploratory data analysis. Later in this chapter, we will go into the technical intricacies and structure of the several picture captioning systems that we are testing. Finally, we go over the mathematical equations of the machine learning algorithm and the study's assessment matrix.

While implementing the study technique stated in Chapter 3, Chapter 4 addresses the finding, observation, and data analysis. This chapter examines the data's nature as well as its metadata. It also goes into the processes required to clean the data. It also discusses the EDA and TTR findings and their implications for the research endeavor.

In chapter 5, we review the outcomes of the different experiments carried out using the concepts and procedures described in chapters 3 and 4. The evaluation measures are then used to contrast the findings and determine the one of the best sets of the models.

Chapter 6 is the thesis's final and concluding chapter, and it contains the study's ultimate conclusion. It begins by using good reasoning to address the research questions presented in Chapter 3. It also discusses the study's addition to current knowledge. This chapter also highlights the constraints and obstacles encountered throughout the study's execution. Finally, we address prospective recommendations that may be implemented in the next phase to solve the issues.

CHAPTER 2

LITERATURE REVIEW

2.1 Visual Encoding

The initial problem about an image translation workflow is to effectively represent the visual content. The visual encoding proposals that are presently can be divided further into few main methods:

1. Non-attentive approaches rely on CNN properties.
2. Methods of incremental attentiveness that encircle the visual information with grids or areas.
3. Visual connections between visual areas are used in approaches based on graph.
4. Approaches based on self-attention that use methods based on area, image-to-text, patch/grid which results in paradigms based on engine.

2.1.1 Global CNN Features

Models which use images as inputs (visual inputs) have an increased efficiency and performance with the usage of CNNS. Image captioning is an application that uses CNNs for better performance. The CNN's last layer can be used in language modelling as it can be used to understand the representations.

In the first article, "A neural image caption generator", this technique is implemented and the output obtained from GoogleNet was given as an input to the language model's first hidden state. Global features that are obtained from the AlexNet as mentioned from Karpathy et al., 2015 consider that as a feed in for the linguistic model. As described by Mao et al., 2015, "Deep Captioning with Multimodal Recurrent Neural Networks (mRNN)" and also from Donahue et al., 2015, it can be noted that they used these features (global) from VGG net at every rate of the linguistic model.

Chen et al., 2015, Fang et al., 2015, Jia et al., 2016, You et al., 2016, Wu et al., 2016, Gu et al., 2017, Chen et al., 2017, Chen et al., 2018, the overall CNN traits happened to be shown in different applications of Image captioning. From Rennie et al., 2017, a Fully Connected paradigm can be created, wherein images are represented using only a ResNet101 yet their original dimensions are conserved. Other approaches by Yao et al., 2017 and Gan et al., 2017 incorporated high-level characteristics (attributes) or tags, that have been modeled over the common words that are taken from the training corpus (captions) in terms of a probability distribution.

Global CNN features have a major advantage, i.e., the simplicity and minimalistic character of presentation which incorporates the strength to obtain and disintegrate whole input's data, but also consider the image's overall context. But the approach of this Fully Connected paradigm makes the content less efficient making the captioning mechanism harder.

2.1.2 Using Attention upon the Grid of CNN Features

Many approaches that are introduced afterwards have boosted the resolution stage of image (visual) encoding, captivated by the limitations of global representations mentioned by Rennie et al., 2017, Xu et al., 2015, Lu et al., 2017. To integrate geometric features explicitly into the language model, Dai et al., 2018 adopted 2-Dimensional activation maps rather than 1-Dimensional global feature vectors. Using the additive attention mechanism from a computational translation literature, a huge portion of captioning has enriched the image captioning design with feature encodings that vary over time by making the model a flexible one.

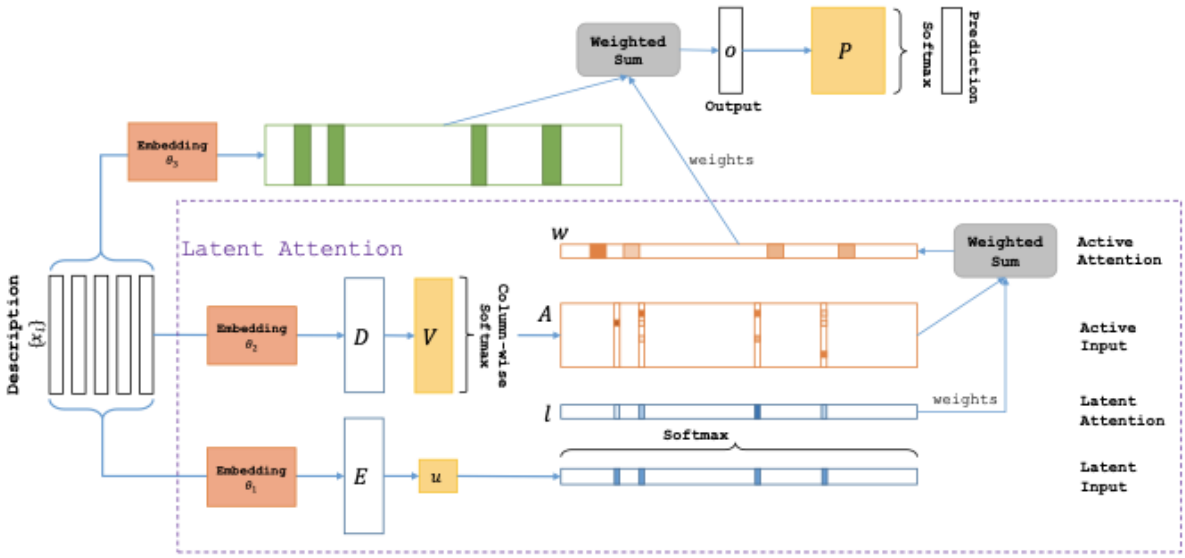

Figure 2.1 Attention in Neural Network

2.1.2.1 Additive Attentions

Attention uses a fundamental premise for calculation called the weighted-average. A single layered feed-forward neural network with a 'tanh' has been proposed by Bahdanau et al. 2014 for sequence alignment. Attention weights are calculated through a nonlinearity.

Consider two sets of vectors which are generic from s_1 to s_n and from t_1 to t_m. The additive attention between the ith vectors of h and x is given as:

$F_{atten}(t_i, s_j) = W > 3*\tanh(W_1 t_i + W_2 s_j)$, where W_1 and W_2 are weighted matrices accordingly, and using the linear combination of W1 and W2, we obtain a weighted vector W3. $p(s_j \mid t_i)$, a probability distribution, is obtained with the help of a "softmax function". This depicts how significant the particular element encoded by s_j given t_i. Although the attention mechanism was actually created to represent the communication between two elements which are sequential, it can be used for a language model by connecting its hidden states with a set of visual image representations.

2.1.2.2 Attention over Convolutional Activations

From Xu et al., 2015, there developed a first technique for leveraging additive attention over a convolutional layer's spatial output grid. By choosing a set of features for every text (word) that obtained, the used model accepts to look-into some sections of the grid thereby extracting the last CNN layer of a VGG net's activation. Upon extracting, it performs attention using additive mechanism to calculate every grid's weight thereby interpreting the relative significance by predicting the next word in the sequence.

Dimensional maps of activation rather than 1Dimensional global feature vectors to introduce the architecture into the language model. Based on the translation literature of machine, the large part of the captioning association has used the attention mechanism.

The insight behind the attention mechanism is to reduce the weighted average. The first formulation was proposed by the Bahdanau et al., 2014 for sequence alignment, a one-layered feedforward neural net having hyperbolic tangent nonlinearity generally used to calculate attentional weights. For example, let us consider two collective group of vectors $\{y_1, .., y_n\}$ and $\{k_1, ., k_m\}$and the additive attention value between k_i and y_j is calculated below:

$$f_{att}\,(k_i, y_j) = v> 3\ tan(k)\ (v_1\ k_i + v_2\ y_j)$$

where v_1 and v_2 are matrices weight and v_3 is a vector weight which represents a straight join. Here one of the functions like soft_max is related to acquire a normal distribution of probability $p\ (y_j \mid k_i)$ which represents the relevance of the element encoded by y_j to k_i. Though the attentional process was originally developed to model the relations between the hidden layer states of a recursive encoder and decoder (two sequences of element), where adaption can be occurred for the connection of a group of visual elements represents the language model hidden states. Participate in convolution activations. Xu et al., 2015 presented the initial method that takes advantage of additive attention models on the geographical output mesh/grid of CNN layer. It let the method to focus selectively on particular components of the grid only by choosing one of the features subset for each and every generated word. This model particularly takes out the activation from the previous layer of VGG net, further it uses additive attention for computation of weight for each grid component, which is transcribed as that element's respective importance for producing the successive one.

Network Review - For example, Yang et al., 2016 added a recurrent review network to the codec framework. It performs a set number of mindful scanning steps in various hidden states that are in the encoder, generating a vector (thought vector) for every possible count. This vector is used in the decoder framework by the attentional mechanism. Multilevel functions:

As proposed by Chen et al., 2017, the channeled attention to CNN geographical attention can be acknowledged. Jiang et al. 2018 go in the same direction. They proposed using multiple CNNs to leverage their supportive information, and then merged their representations using a recursive technique.

Normalized fixation histograms over the input as image module of Xu et al., 2014 and weighting visited parts are supported if they are fixed or not. Lines from Tavakoli et al., 2017, Ramanishka et al., 2017, Cornia et al., 2018 used salience maps for sorting.

2.1.3 Attention to Visual Regions:

The intuition for using highlighting comes from neuroscience, suggesting that our brains integrate a pragmatic thought of using visual signals in a bottom-up approach. Inputs that leverage our awareness, while the disordered flow provides visual catalyst that adjust previous predicted targets.

2.1.3.1 Attention using Bottom-up and top-down approaches

Now it's then combined with a pragmatic process which weigh each predicted word with the respective region. It uses Faster RCNN by Ren et al., 2015 for detection objects, resulting in a vector of pooled features of every suggested region. It let the prediction of model a rich, large set of recognitions, including labelled objects, and encourages to learn the feature representations in a better way.

2.1.3.2 Other proven approaches

For years, the image annotations use a standard set of visual image functions in order to deal with the provided raw input. Using this strategy, a few of the mentioned works used to build the image coding wave such as, Ke et al., 2019, Qin et al., 2019. The two notable mutations, Visual Guideline: While the typical image attention points on a whole image region at every

step reported by Zha et al., 2019 proposed approach, establishes a network of sub-guidelines that also interpret by encoding visual actions to work for as context for the following activity through a Long Short-Term Memory.

Dimensional transformations—Pedersoli et al. 2017 proposal done by using spatial transformers to generate specific areas of by relapsing region suggestions in a supervised manner. In particular, a location net learns a transformation of each feature map location and then uses bilinear interpolation for backtracking the feature vector of every region.

2.1.4 Encoding based on graph

For improving the coding of picture parts & the relations, should examine using graphs built on the image parts to enrich the presentation. The initial attempt in this direction goes back yao et al., 2018 with Guo et al., 2019 proposed to combine both the spatial and the semantic association between objects. The graph of semantic relations is given by the application of a pre-trained classifier in Visual data that anticipate an interaction between various duos of objects. Instead, a graph of spatial associations is obtained by geometric measurements with an understanding on modeling the semantic relationships, Yang et al. 2019, put forward to combine the previously learned denotations from textual data to visual coding by utilizing a graphics – which is centered on representation of images and sentences.

2.1.4.1 H – Trees (Hierarchical)

The whole image is considered as a root whereas the medial nodes depict the image areas and the leaves of the tree depict the objects disintegrate into the regions. Encodings based on graphs provided a mechanism to exploit the relationships among recognized objects, allowing the exchange of data in neighboring nodes and hence locally. In addition, it enables seamless integration from outer cautious information. On the other hand, the manual construction of the structure of graph limits the relationships among the visual features.

2.1.5 Encoding based on Self-Attention

Self-attention is an attentional system in which each and every member of a collective group is interconnected with all others and is used to provide a more efficient design of the assemblage of elements by remaining threads (connections). From Vaswani et al., 2017, it is

reported initially that for computational linguistic translation and comprehension situations, resulting in the Transformer topology and its versions, that have since come to dominate the field of NLP and later computer vision.

Yang et al., 2019 employed a module which is conscious to encode interconnections among features originating from an object detector, out of all the image-captioning models that used this stated strategy. Other research proposed picture captioning-specific improvements to the self-attention operator, Herdade and colleagues, 2019. Guo et al., 2020, Pan et al., 2020; Huang et al., 2019. Furthermore, He et al. 2020 proposed a spatial graph transformer which gather in various types of spatial relationships between recognitions.

2.1.5.1 A over A: (A refers to attention)

Huang et al., 2019 presented an augmentation of attention which is operational in which a context driven gate weighs the final information. In specific, the self-care output is concatenated with the inquiries, followed by the computation of a piece of data and a gate vector, and lastly their multiplication. We employed a method in the encoder to improve visual aspects. Subsequently, X Linear Attention – Pan et al. 2020 approved suggestion using bilinear clustering approaches to increase the representational capacity of the exit function which utilized this strategy. In particular, this process encodes interactions, resulting in an improved group of visual attributes.

2.1.5.2 Memory increased attention:

From Cornia et al., 2020, a framework based on transformers in which the self-attention operator in each and every stage (layer) of the encoder is supplemented with a set of storage coordinates (vectors). In simple terms, new "slots" gained during the training process that may encode multilayer visual connections extend the collection of keys and values. By the reference to Ji et al., 2021, it is advocated that a vector which is accessed globally is determined by considering the feature vectors' average and to be added to the series of feature vectors to boost self-attention. For each layer, a global vector is produced, and the resultant vectors (global) are appended using an LSTM cell to provide a cross-layer representation. From Luo et al., 2021, a proposal that states a hybrid strategy that leverages the combined benefits of the region as well as the grid properties is acknowledged. Each function class

receives two self-attentive modules, and a cross-attentive module brings their interactions together locally.

Liu et al. 2019 suggested an attention module-based architecture to match grids or recognition components with visual feature words retrieved from an idea extractor and generate sound encodings semantically. Aside from the attention operator's application to detections, the importance of lattice features has lately been reevaluated. Transformer-like structures may also be directly applied to tiles, obviating the need for the convolution operator as mentioned by Dosovitskiy and Touvron. Liu et al. 2021 built the first no-convolution caption design in this area. Following that, the same visual coding method was used in CLIP by Radford et al., 2021, and SimVLM by Wang et al., 2021, with the exception which the visual coder was trained from scratch on big noisy data. Shen et al., 2021, Mokady et al., 2021, and Cornia et al., 2021 all employed CLIP-based features in their later captioning algorithms. Self-attention is also used in other works to encode visual information.

Tan et al., 2019, Lu et al., 2019, and early fusion methods were all instrumental in achieving these extraordinary results. Li and colleagues, 2020; Zhou and colleagues, 2020. For example, from Zhou et al., 2020, the encoder-decoder model is integrated into a single transformer layer channel, with the range and text (word) tokens initially merged to become a single stream, which is also stated by Delvin et al., 2018. This is the first time a unified model has been instructed on a significant number of caption pairings and subsequently refined/finely tuned to handle bi-directional and stream-to-stream tasks that involve prediction. In a similar hypothesis, from Li et al., 2020, advocated OSCAR, a topology (architecture) similar to BERT with tags for objects serving as reference points to help with semantic alignment pictures and textual data. Distinguish triple regions of aligned wordmarks from the contaminated. Hu et al. 2021 increased the scale of the VinVL model and used pretraining in this situation.

2.2 The Language Models (LMs):

Initially, the main agenda is to find out the occurrence of a chance (P(x)) that a certain sequence of words will appear in a decision. The image LM part labelling algorithm assigns a certain probability to a sequence of words. Please note that the language model is provided under a condition with respect to the visual encoding. It's worth noting that the language model

finds the next word in an autoregressive fashion, that means each expected word is dependent on the preceding ones.

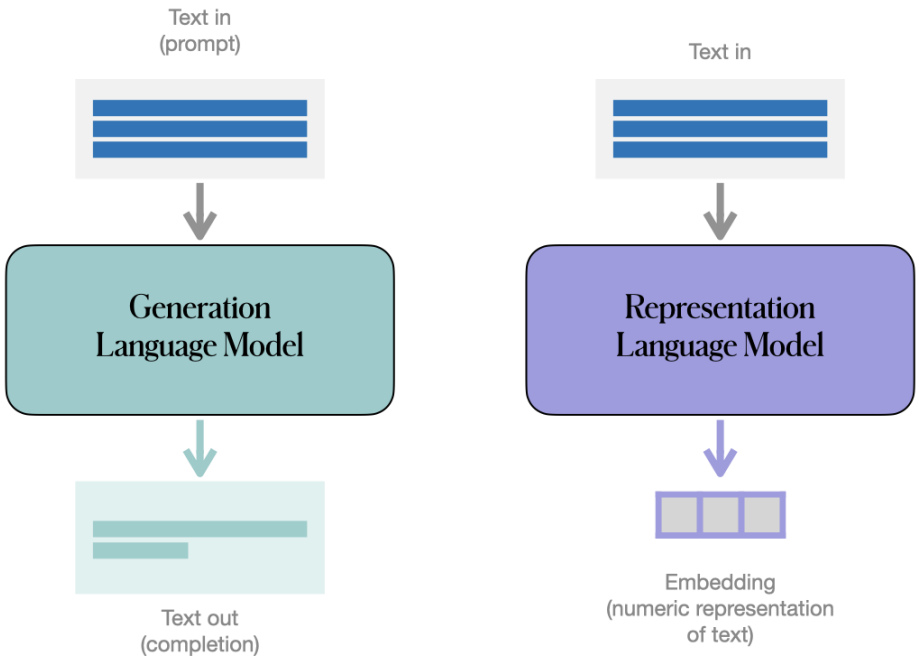

Figure 2.2 Generation & Representation of language model

By sending a particular end-of-stream token, the model normally chooses when to cease creating subtitles. The language modeling strategies applied to captions divided into: which may be monolayer or bilayer; The methods based on CNN that represent a first attempt to overcome the fully recurring scenario; Fully attentive based on transformers Approaches; Early text and image fusion strategies (BERT type) that directly combine visual and textual inputs. Models with LSTM RNNs are well equipped to deal with sentence creation because the language has a sequential structure. Among the RNN forms, LSTM by Hochreiter el al., 1997 was the dominant option for language modelling.

2.2.1 The Single-Layered LSTM

The simplest LSTM-based captioning design, suggested by Vinyals et al., 2015, is based on a single-layer LSTM. The output title is created from the LSTM's initial hidden state, which is the visual encoding. For every step in the time, the hidden state gets projected onto a vector of size equal to the vocabulary

At each step in a time stamp, a word gets predicted by projecting the inner state (hidden layered state) onto a vector the same size as the vocabulary using the S-activation function (Softmax function). During the training of the fundamental truth proposition, input words are taken, whereas during speculation, the input texts are those created in the previous stage. The additive attention mechanism shortly after presented by Xu et al., 2015. The context of the target word and also for the prediction of each sense of the target word, context vector is created and fed to the MLP that predicts the output word if the prior inner state directs an attention process to visual attributes X.

2.2.1.1 Other approaches

Yang et al., 2016, Chen et al., 2017, Pedersoli et al., 2017, among others, adopted a single-layer LSTM-based decoder, largely without spatial alterations, while some offered major modifications. Lu et al. 2017 expanded the geographic picture attributes with an extra learning vector, the visual sentinel, that the decoder provided instead of the visual features when "non-visual" words ('of', 'the', and 'in') are being formed, which doesn't require any visual characteristics (features). The visual character at a checkpoint is computed and created from the preceding concealed state word at each time step. The model will then build a context vector from the supplied picture and visual sentinel features, with the relevance of each feature weighted by a learnable gate.

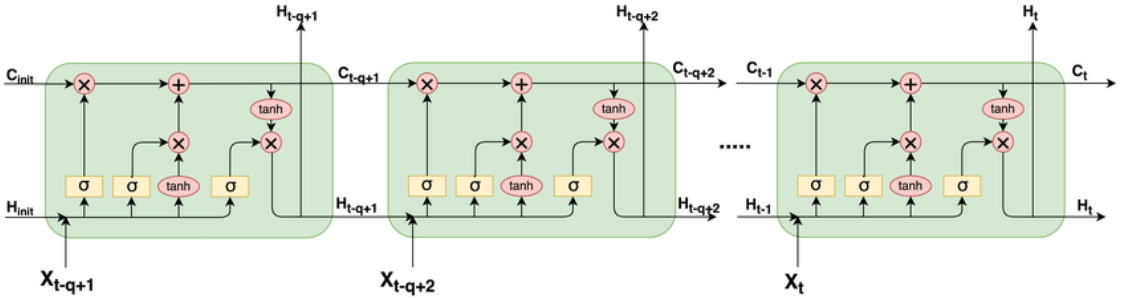

Figure 2.3 Single layered LSTM Architecture

2.2.1.2 Reconstructing the Hidden state

The main idea is to use a second LSTM cell in order to recreate the previous (past) hidden state. This idea is proposed from Chen et al., 2018 where the previous h-state is dependent on the present state as it is advantageous in applying model's dynamic transition. In order to

improve the context modelling, a helper module is employed with two bidirectional LSTM cells as advocated by Ge et al., 2019. The additional supplementary module from one direction, approaches the h-state of the LSTM in the opposite address. To generate the result (caption), an “Intermodal-attention” mechanism integrates the grid's visual elements with the duo set of bidirectional LSTM cells.

2.2.1.3 Multistep generation

Wang et al., 2017 propose separating the caption creation process into two steps to get from a rough pith to fine qualities: Single-layer LSTM was used to implement skeletal set creation and attribute enrichment. One of the researchers, a multi-level coarse-to-fine framework was built using a sequence of Long Short-Term Memory decoders, each of which worked on the output of the preceding one to produce progressively complex subtitles. LSTM with semantics – Jia et al., 2015 developed an LSTM extension to guide the development of semantic information retrieved from the picture. This mantic information is specifically employed as an extra input for each LSTM gate.

2.2.2 The Two-layered Long Short-Term Memory

To strengthen their capacity for apprehending the higher-order interactions, Long Short-Term Memories can be expanded to multiple -layers of structures. Donahue et al., 2015 introduced the first two-layer Long Sort term memory which takes as subtitle language model, assembling 2 substrates, that the input of the later are the hidden states of the former one. He suggested that the 2 layers be specialized to execute visual attention and real language modelling. Using the formerly generated word, the prior concealed state, and the image's pooled mean attributes, the initial layer of Long Sort term memory works as a top-down visual attentional model. Using an additive attention method, the present hidden state is then utilized to build an occurrence over the picture areas. The next layer of Long Sort term memory receives the vector of image attribute and combines it with the hidden state from the initial layer to generate the probability of distribution across the vocabulary.

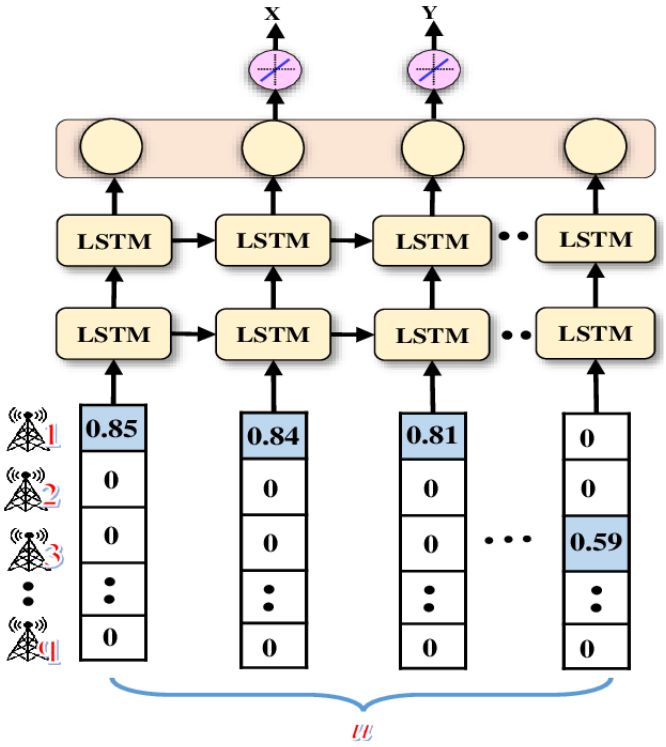

Figure 2.4 Bi - stacked Long Short-Term Memory

2.2.2.1 Two-layered Long Short Term Memory variants

Because of its representational strength, LSTM was the most widely used language model technique prior to the development of transformer-based architectures, as indicated by Yao et al., 2018, Yang et al., 2019, Shi et al., 2020, and Yao et al., 2019. As a result, several alternatives have been presented to improve the performance of this strategy. Lu et al., 2018 used a signaling network to control the attentional process based on content. The network anticipates holes in the label during the creation process, which are then filled with image area lessons. On non-visual words, a visible guard is worked as a false floor. The object detector is used as a feature region extractor as well as a word pointer to the model in this technique. The attention which is reactive — proposed two reflexive modules Ke et al., 2019: the first analyses the connection among the internal states of all anticipated words in the past and the present word, while the second enhances syntax sentence structure by managing the process caused by words sharing positional information.

2.2.2.1.1 Backpropagating and Predicting

Similarly, two modules were employed by, Qin et al., 2019: the retrospective unit, that takes help from the previously serviced vector to calculate the successive one, and the direct prediction engine, which predicts the two new words at the same time, reducing the problem of assembled errors that can takes place during intrusion. Duration for Attention adaption: An adaptive attention time technique was introduced Huang et al., 2019 where the decoder may take a random number of attention steps for every produced word, which is set by a trusted network on the second LSTM layer.

2.2.3 Strengthening Long Short-Term Memory having Self Attention

Huang et al., 2019, Pan et al., 2020, Liu et al., 2020, and Zhu et al., 2020 used the self-attention promoter instead of add-on attention in LSTM-based language models. Huang et al. 2019 specifically expanded the LSTM with the focus on Attention operator, which calculates an attention in addition. The X Linear attention block was developed Pan et al., 2020, which inflates self-attention through 2^{nd} ordered interchanges and upgrades both visual encoding and the language model. Zhu et al., 2020, on the other hand, use the architecture of neural network for searching the prototype to pick the relations among the substrates and the working within the gates of RNN-based language models for captions using a self-attention enhanced decoder. A noteworthy method is that of Aneya et al., 2018, which use convolutions as a language model. A global picture feature vector, in particular, is mixed with word embeddings and supplied to a CNN, which analyses similarity among all the words during training and sequential inference. To prevent the model from exploiting future word token information, convolutions are correctly disguised.

2.3 Transformer-based Architectures

Vaswani et al (2017).'s completely aware paradigm was proposed. It profoundly altered the way people thought about language formation. Shortly after, the Transformer model provided the foundation for further developments in NLP, such as BERT by Devlin et al., 2018 and GPT by Radford et al., 2019, as well as the widely accepted architecture for many languages processing tasks. Transformer architecture was also employed. A masking strategy is used to

the preceding one word during training to confine a one-way generating process. Some closed caption models used the original Transformer decoder with minor design changes. Furthermore, strategies for increasing voice output and coding of visual feature activation methods have been created.

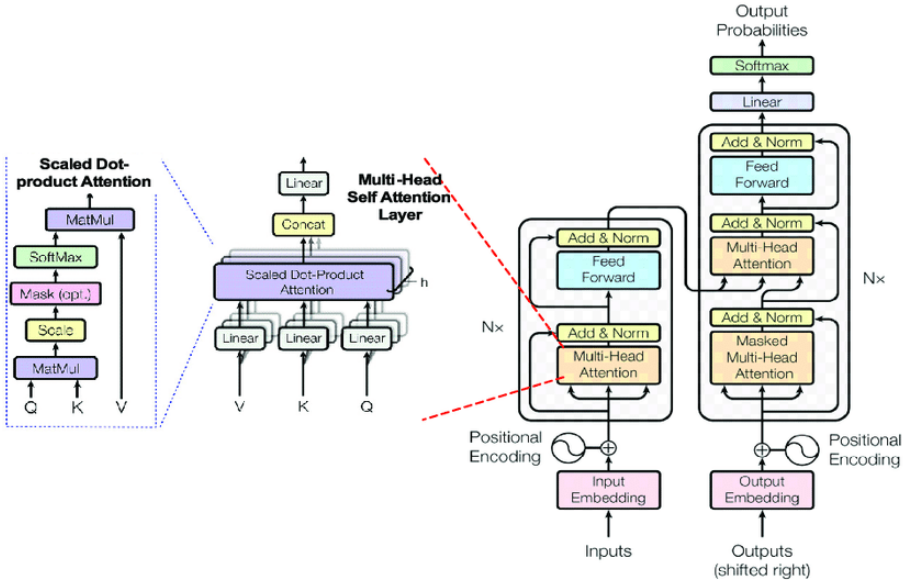

Figure 2.5 Transformer based architecture

By merging and modifying image-space characterization with semantic properties by an external tagger line, Read et al., 2019 suggested a triggering procedure to the cross-attention operator that regulates the flow of visual and semantic information, a context gate mechanism will incorporates by Ji et al., 2021, to adjust the effect of the global picture representation on each produced word, which is represented by attention that is multi headed. Cornia et al., 2020 proposed taking into account all levels of coding rather than only crossing on the final stone. They created the mesh decoder to do this, which comprises a mesh operator that modulates the contribution.

Although the encoder-decoder paradigm is widely used in picture annotation, some work has altered the annotation structure to employ a BERT-like structure from Devlin et al., 2018, the visual and textual modalities given in merge with the initial stages. The fundamental advantage of this design is that the text-processing layers may be pre-trained using parameters learnt from enormous textual data. As a result, the BERT concept is commonly employed in works that make advantage of past training. The first example comes from Zhou et al., 2020,

who created a unified model for captions that combines visual and textual modalities in a Bidirectional Encoder Representations from Transformers - like design. This is composed by sharing for both encoding and decoding by multilayer Transformer-Encoder network that has been pre-trained on a large corpus of picture label pairs and then tailored for image labels by masking the sequence of tokens to emulate the one-way creation process. Furthermore, Li et al., 2020 proposed using identified item labels in images as anchor points to achieve better alignment combined representations of sight and language.

2.4 Discussion

Varied types of techniques have been discussed in the literature review that are implemented for image captioning in languages apart from English. We have understood how the image captioning works by categorizing the whole application into visual encoding and language model sections. Starting from how an image can be vectorized to how a language model can be built, we have spent a remarkable time on the research.

The main aspect of this research is to understand how an image can be decoded into Telugu language. So, we understood how to build a language model and build word vectors for the language with the help of various research applications provided in this Literature Survey. We understood the usage of pre-trained word embeddings for Telugu language provided by ‘fastText’, which also provides vector embeddings for various Indian Languages.

2.5 Summary

In this chapter, we thoroughly discussed the topics and the methodologies that are implemented in the field of image captioning. The aims and objectives that are mentioned in the chapter-1 are based on the methods learnt from the Literature Survey. We discussed how the visual encoding is done and moved to language model building for Telugu language (Indian languages).

We focused on obtaining an efficient application with basic architectures (Encoder-Decoder) and studied various applications based on the same architecture with the usage of various pre-trained models.

CHAPTER 3

METHODOLOGY

Our model uses a generic method for training. For acquiring great results, and to apprehend extensive details, the larger datasets are to be used. In this project, we used smaller Flickr 8k dataset which contains of 8k images (8,000). The five different captions are annotated for each image. All the captions are in English itself. Using Google Translate API, these captions are translated to Telugu language. Inception V3 model is pre-trained on the image net dataset which runs by our model.

Preprocessing is done for every image of the dataset by this model and reshaping is done for the output features in custom dimensions and gets stored. There are some undesired characters for the captions of each image which were cleaned parallelly. For the training words, punctuations are avoided for the model. Subsequently, tokenization is done for the sentences which adds the special tokens at the starting and the ending to signify the same. The captions which are modified and generated are trained on RNN of LSTM structure. By taking help of word embeddings of Telugu which are pre-trained fastText, the modified images are directed on a custom neural network which followed by the captions.

The model is usually used on the input of an image to predict the successive similar word and its captions which were partial. For prediction purpose, trained weights are used and get saved. The image of a word and its previous related words have been calculated with the probabilities.

3.1 Model Structure

The encoder-decoder neural network model has developed, it is a most important thing to choose suitable model structure. However, so many researches are there to select best model for given task. 16 different model structures for image captioning and also identifies the best model. In this project we had make use of Merge model and it is depicted below. The Merge

model includes two vectors as input Image feature vector and word sequence vector. The combination of these two input vectors will be used to generate next word in the sequence using the decoder model where text and image data handled to perform better. To encode data, RNN – LSTM neural network model is used to generate image captions.

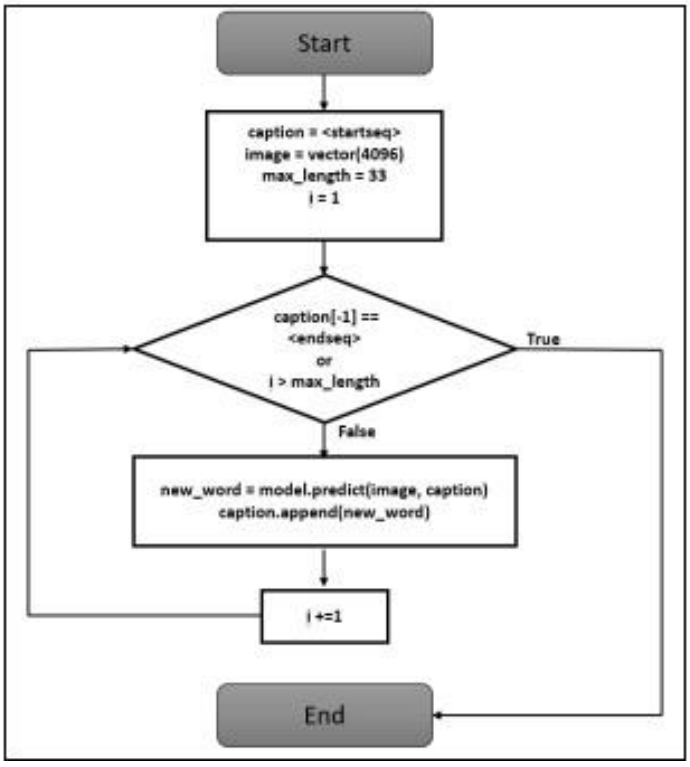

Figure 3.1 design process flow image captioning model

3.2 Dataset

Flickr 8k, Flickr 30k, Microsoft COCO. are some of the various large and abundant datasets used for image captioning. Out of all these models by keeping in view of their limitations in time and in computing, Flickr 8k dataset was proved to be the mostly suitable model. It's a dataset that consists of 8k images which are random general-purpose images and they are provided with five annotations each. By using google Translate API, each and every annotation is translated into Telugu language. The previous translations were marked with imperfection due to the backend issues of API which we're currently using (Google Translate). Few annotations were not as expected due to limited iterations. But these

imperfected issues can be overcome using a library that can identify defects of the annotations which are translated. Iterations have been done for all the captions by Google Translator API and for verification we have used an API that can detect defects.

3.3 Convolutional Neural Networks (CNNs)

In order to acquire image features, we made use of CNN. Basically, the arrays of resized image drop the multiple convolutional layers which leads to production of feature vectors. For every image that is provided, the generated feature vectors are always of static. For achieving good accuracy, we need to perform large amounts of fine tuning. However, for usage of pre-trained models on wide range of globally used datasets is preferred. Availability of CNN models are many to say. Few of them are AlexNet, Resnet50, VGC16, Inception V3. Some of them have used to get the best model.

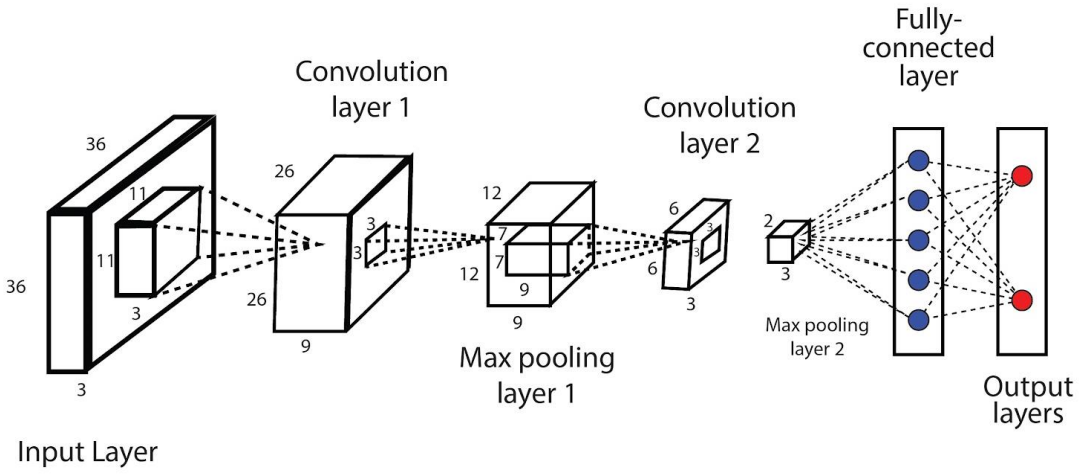

Figure 3.2 CNN Architecture

Out of all, AlexNet was the first Deep Convolutional Neural Network which acquired 84.7% accuracy on dataset (ImageNet) LSVRC – 2012. Whenever the Vanish Gradient problem is arised, we applied this model containing ReLU as the activation function. ReLU, activation function is used mainly to solve Vanish Gradient. As proposed by K. Simonyan and A. Ziseman, VGG16 which acquired the better test accuracy in order to perform visual classification on 1K various features by ImageNet dataset. Compared with all other models, Inception V3 has less parameters which was chosen.

Pre-trained model	Input Parameters	Accuracy in Top - 5
AlexNet	62 Million	85.2%
VGGNet	138 Million	91.8%
Inception	6.4 Million	94.0%

Table 3.1 comparing all the CNN models

From the above table, Inception V3 has highest top 5 accuracy. This model accuracy comparison leads to go for Inception V3 which used ImageNet dataset to be trained in order to perform over 1K different features by image classification. For every image, there is an extraction of 2048 length vector, the last layer of CNN which is famously known as Softmax layer was removed.

3.4 Recurrent Neural Networks (RNNs)

In order to understand the relation between various words of various sentences that occur in different time series, Recurrent Neural Networks (RNN) are employed. Exploding Gradients & vanishing Gradients are the two major problems in RNN. Glavnoye Razvedyvatelnoye Upravlenie and Long Short-Term Memory are used to solve this kind of problems.

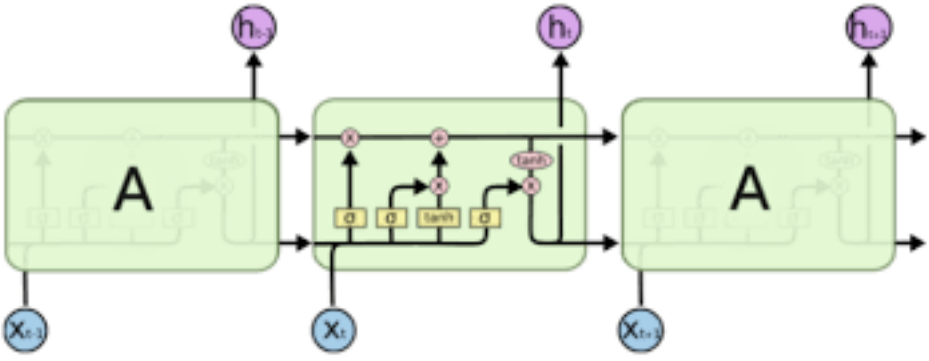

Figure 3.3 Inception V3 Architecture

The main problem faced by RNN is the long-term dependencies. To reduce this problem long short-term memory is a type of RNN which can be used. The chain like structures are LSTMs which is shown in the figure. Another latest trending type of RNN is gated Recurrent units (GRU) which are similar to LSTM. Hidden states are used instead of the cell states in GRU. It's a very simple and basic architecture GRUs are used to build bigger networks but there will be more flexibility in LSTM. LSTMs are significantly proven than GRUs. So, we have opted to use LSTM by considering the above reasons.

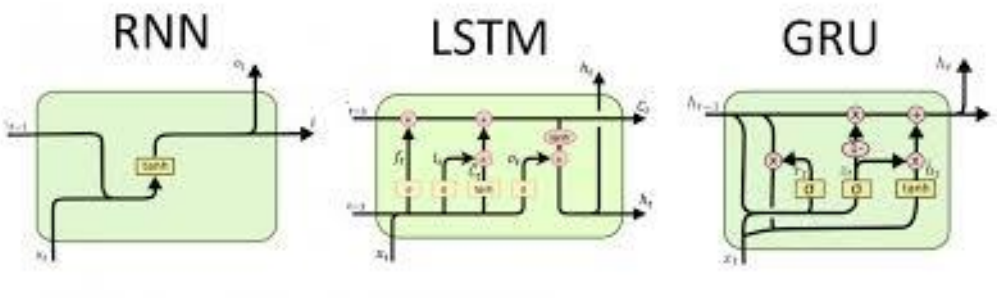

Figure 3.4 RNN, LSTM, and GRU Architectures

3.4.1 Word Embeddings

Machine Learning and Deep Learning algorithms cannot analyze text independently. They execute many tasks but rely on statistics to achieve excellence. To process any text in any language, we must be fluent in that language. To execute any activity, models must comprehend the relationship between certain terms. In our project, we need the necessary data so that our model can interact with words to generate grammatically correct phrases. To tackle this problem, word embeddings are used to represent words as integers so that they may be utilized for language modeling and feature learning approaches in Natural Language Processing.

The fixed dimensions of the words give the features and it helps to know the similarities in words. There are several techniques like fastText, GloVe, Word2Vec. All these methods will work differently to produce same word embeddings which contain several features.

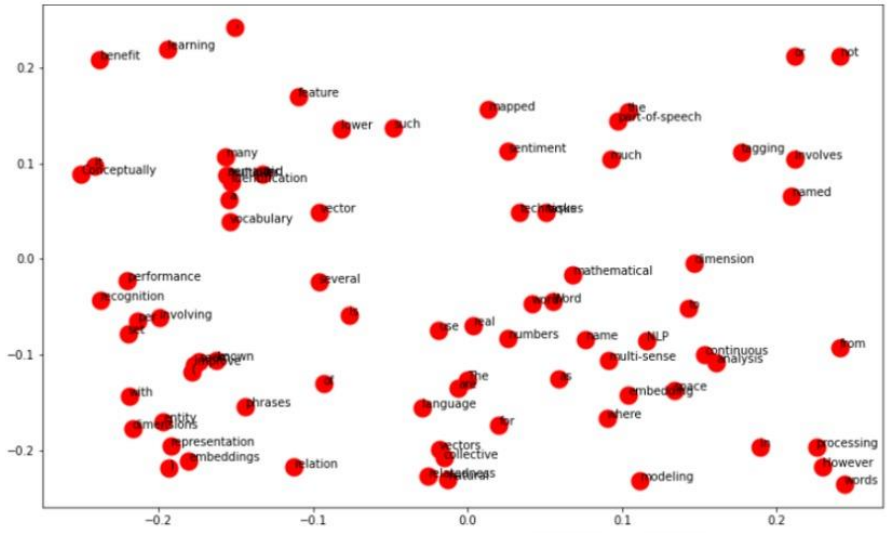

Figure 3.5 A visualization of word embeddings

Word Embeddings for Telugu language are the major requirements for our project. To create a single word embedding needs a high-quality standard. It's a time-consuming task which make us towards the different pre-trained word embeddings. There were no good pre-trained word embeddings. There were no good pre-trained word embeddings in any Indian Language. Thirteen Indian Languages for the word embeddings are provided. Alternative word embeddings file in Telugu for 300 dimensions was found. Every feature represents a word and shows every dimension. There is still a requirement for refining from native speaker. It was best Telugu word processing so we proceeded with it.

3.4.2 Architecture of the model

3.4.2.1 Image Processing

For converting into Telugu language, our project needs word embeddings. A wide range of high-quality standards are essential to create one task which needs to be completed requires a lot of time, so to complete the task in less time, we used trained which is previously done word embeddings in our project. Good pre-trained words were not found for any Indian Languages. Word embeddings for thirteen Indian languages are provided. Languages

translated to Telugu were the downside. A fastText word embedding file were found as a substitute for Telugu with three hundred dimensions. Each dimension considers each individual feature with their specific word. Despite of the fact that the Telugu word embeddings are found, there is a necessary rectifying from the outlook of the native speaker. The Telugu word embeddings which we found were the best. So, we have to move forward with those embeddings.

3.4.2.2 Caption Processing

Here we take starts token consisting of half/partial caption that implicated starting of the caption by making it zeros to match the largest caption length by doing it fixed size that given to an embedded layer which is loaded with fastText word embeddings. These are the individual words of captions which is mapped to its respective embedding vector of 300D. Subsequently, it is over on to dropout layer follows by Long-short term memory layer consisting of 256 neurons.

3.4.2.3 Combined Processing

The outputs of the processed images and partial caption are a 256-dimensional vector of identical size. The combined input is jumbled with a hidden Dense layer of 256 neurons with the ReLU activation function that was taught by the model from the extra layer. The length of the vocabulary vector is the same as the output dense layer, which consists of neurons with Softmax activation for predicting the probability of each word. The model is constructed using the categorical cross-entropy loss function and Adam optimizer.

3.5 Training

Dataset contains image titles and its five captions for every image, but the requirement of image with partial captions must be shown as result for the next possible word outcome by our models. Padding is required to get similar size because every partial caption length should be as same as the longest caption. Due to computational limitations all the calculations were

showed slow. Using 100 epochs training was done on the model with a certain step size which equals to number of training images and divided by number of images set (per batch).

For the next 50 epochs of double and triple as step size was considered and makes 200 epochs as total count. There will be a huge of experimentation before firing the epochs and step size computational hardware requirements which are used for deciding parameters of training are the major limitations.

Using unique model structure and various parameters several models are implemented in the experiments. Here, the present section clearly explains the structure of the final model which has given high-quality image caption. Using minibatch stochastic gradient descent method with fixed learning rate and no momentum, the model was trained and there was computed by back propagation. Except for the image encoder weights were initialized randomly due to the pre-trained CNN usage in encoder model. The structure of fine-tuned image captioning model is explained as:

3.5.1 Image Feature Input

The input of the image encoder model is a 4096-element image feature vector that was created using VGC-16. Therefore, the input shape of the image encoder is configured to match the size of the picture feature. The dropout layer is inserted with a dropout rate of OS, and the dense layer's image feature vectors are reduced to 128 elements using the ReLU function.

3.5.2 Text Feature Input

As a second input, the text encoder model receives the description of a picture. There is an assumption that the input phrase will have a maximum length equal to the length of the description given via the encoding layer's mask, which ignores padding values. It is then followed by a layer with a dropout rate of 0.5. In the last step, an LSTM layer with 128 memory units is used to generate 128 element vectors.

3.5.3 Merge and Prediction Output

Input was taken from image encoder model and text encoder model by the decoder model. 128 element vectors are produced by both the encoders and merged using an addition

operation. Using 128 neurons it will be forwarded to dense layer. At final stage, dense layer makes Softmax prediction of the word.

3.6 Summary

In this chapter, the research approach was explained in depth. The various concepts we covered in this chapter are dataset collection, data analysis and cleaning, data pre-processing, encoder-decoder mechanism, training processes and algorithms.

The data analysis, cleaning, preprocessing and visualization took the most time as this is an essential job that has to be taken care of before proceeding to the model building and training. We analyzed if the data is imbalanced/normalized and checked if imputing is needed.

We built an encoder-decoder model and analyzed by using various pre-trained models (CNN/RNN) which will be discussed in the analysis section.

CHAPTER 4

ANALYSIS

4.1 Introduction

In the preceding chapter, we reviewed the research approach used for this study. This chapter discusses the research's analysis and conclusions while executing the approaches. The first

part discusses the dataset, its cleaning and preparation, followed by visual encoding using CNNs, and lastly language models or decoding models using RNNs.

4.2 Dataset

In this term paper, we have employed the flickr8k dataset. This dataset is accessible on Kaggle in CSV format. This data set comprises 8000 images with 32000 Telugu-language descriptions. The data set has been thoroughly cleaned and includes training and test sets that may be used to assess Telugu classification algorithms. The dataset includes two geographic regions:

- images: - It contains ~8k images which ought to be classified and captioned.
- captions: - Each picture contains five conceivable combinations of captions in Telugu which in add up to is ~32k.

Of the 40454 information columns, 32364 are used for preparing data, and 8090 are used for testing the model. Due to the absence of partitioned validation datasets, stratified random sampling was used to generate validation datasets. This is important to ensure that the test population is representative of the whole population and that each subgroup is addressed as necessary. We created a 3500-row validation dataset using this method.

image	caption
1000268201_693b08cb0e.jpg	ఒక గులాబీ దుస్తులలో ఉన్న పిల్లవాడు ఎంట్రీ మార్గంలో మెట్ల సమితిని అధిరోహించాడు.
1000268201_693b08cb0e.jpg	ఒక చెక్క భవనం లోకి వెళుతున్న ఒక అమ్మాయి.
1000268201_693b08cb0e.jpg	ఒక చిన్న అమ్మాయి ఒక చెక్క ప్లేహౌస్ లోకి అధిరోహణ.
1000268201_693b08cb0e.jpg	ఒక చిన్న అమ్మాయి ఆమె ప్లేహౌస్ కు మెట్లు పైకి.
1000268201_693b08cb0e.jpg	ఒక పింక్ దుస్తుల లో ఒక చిన్న అమ్మాయి ఒక చెక్క క్యాబిన్ లోకి వెళుతున్న.
1001773457_577c3a7d70.jpg	ఒక నల్ల కుక్క మరియు మచ్చల కుక్క పోరాడుతున్నాయి
1001773457_577c3a7d70.jpg	ఒక నల్ల కుక్క మరియు ఒక ట్రై-రంగు కుక్క రోడ్డు మీద ఒకరినొకరు ఆడటం.
1001773457_577c3a7d70.jpg	ఒక నల్ల కుక్క మరియు గోధుమ మచ్చలతో ఉన్న తెల్లని కుక్క వీధిలో ఒకరికొకరు చూస్తారు.
1001773457_577c3a7d70.jpg	రోడ్డు మీద ఒకరినొకరు చూస్తున్న వివిధ జాతుల రెండు కుక్కలు.
1001773457_577c3a7d70.jpg	పేవ్మెంట్ మీద రెండు కుక్కలు ప్రతి ఇతర వైపు కదులుతాయి.

Figure 4.1 Dataset

4.3 Data Preparation and Analysis

In this section, we undertake the data cleaning operations necessary to generate a dataset fit for use as input in word embeddings. As outlined in Chapter 3 of Research Methodology, we are executing the following data cleansing procedures. Removal of uncommon(special) characters

- Elimination of English characters
- Removal of digits
- Deletion of extra spaces
- Removal of stop words
- Take out all single characters

Be that as it may, we have kept utilize of the block-words and the cleaning strategy constrained to the word embeddings. After doing EDA on the dataset, we found the there are a significant amount of blocking words. We used block-words provided by the NLTK library.

caption	block_words	block_words_count
ఒక చారల చొక్కాలో ఒక యువ బాలుడు ఒక చెట్టుకు వ్యతిరేకంగా వాలుతున్నాడు, అయితే మరొక చైల్డ్ ఒక పిక్నిక్ టేబుల్ వద్ద ఉంటుంది.	{వ్యతిరేకంగా, మరొక, ఒక, వద్ద, అయితే}	5
ఒక బూడిద చొక్కా మరియు గోధుమ జుట్టు లో ఒక అమ్మాయి కుక్క చుట్టూ కట్టుబడి గా పచ్చిక మీద నడలింపు.	{మరియు, చుట్టూ, గా, ఒక}	4
ఒక ఎరుపు దుస్తులు ధరించి ఒక అమ్మాయి మరియు ఒక పైరేట్ ధరించి మరొక అమ్మాయి చుట్టూ ప్లే.	{మరొక, మరియు, చుట్టూ, ఒక}	4
సైనికులు 2 మరియు పౌరులు ఒక బస్ స్టాప్ వద్ద ఒక బెంచ్ మీద కూర్చొని ఉన్నారు	{ఉన్నారు, వద్ద, మరియు, ఒక}	4
ఒక టాన్ కుక్క ఒక లంగా మరియు ఒక నలుపు మరియు టాన్ కుక్క దూరంగా వాకింగ్ ఒక మహిళ వద్ద ఎగరడం.	{వద్ద, మరియు, దూరంగా, ఒక}	4
ఒక పసుపు స్నానం సూట్ లో ఒక అమ్మాయి నవ్వుతూ మరియు ఒక నారింజ దావాలో ఒక అమ్మాయి వద్ద పాయింట్లు, మరొక అమ్మాయిలు కనిపిస్తోంది.	{వద్ద, మరొక, మరియు, ఒక}	4
ఒక మనిషి రెండు బెండాలు మరియు ఒక డ్రాగన్ ముసుగు మోసుకెళ్ళే మరొక వ్యక్తి మధ్య నిలుస్తారు.	{మరొక, మరియు, మధ్య, ఒక}	4
ఒక మహిళ ఎర్ర టోపీ మరియు ముఖం పెయింట్ మరొక మహిళ వద్ద నవ్వి.	{వద్ద, మరొక, మరియు, ఒక}	4
బొమ్మలు మరియు మరొక బాలుడు చుట్టూ ఒక ఆకుపచ్చ చొక్కా ఒక బాలుడు, పెంగ్విన్స్ తో ఒక నీలం చొక్కా ధరించి, తన ఎడమ వైపు	{మరియు, మరొక, చుట్టూ, ఒక}	4
ఒక బాలుడు మరియు ఒక డాక్ మీద ఒక కుక్క మరొక కుక్క దూరంగా ఈత చూడటం.	{మరొక, మరియు, దూరంగా, ఒక}	4

Figure 4.2 Count of block words

caption	block_free
ఒక గులాబీ దుస్తులలో ఉన్న పిల్లవాడు ఎంట్రీ మార్గంలో మెట్ల సమితిని అధిరోహించాడు.	గులాబీ దుస్తులలో ఉన్న పిల్లవాడు ఎంట్రీ మార్గంలో మెట్ల సమితిని అధిరోహించాడు.
ఒక చెక్క భవనం లోకి వెళుతున్న ఒక అమ్మాయి.	చెక్క భవనం లోకి వెళుతున్న అమ్మాయి.
ఒక చిన్న అమ్మాయి ఒక చెక్క ప్లేహౌస్ లోకి అధిరోహణ.	చిన్న అమ్మాయి చెక్క ప్లేహౌస్ లోకి అధిరోహణ.
ఒక చిన్న అమ్మాయి ఆమె ప్లేహౌస్ కు మెట్లు పైకి.	చిన్న అమ్మాయి ఆమె ప్లేహౌస్ కు మెట్లు పైకి.
ఒక పింక్ దుస్తుల లో ఒక చిన్న అమ్మాయి ఒక చెక్క క్యాబిన్ లోకి వెళుతున్న.	పింక్ దుస్తుల లో చిన్న అమ్మాయి చెక్క క్యాబిన్ లోకి వెళుతున్న.
ఒక నల్ల కుక్క మరియు మచ్చల కుక్క పోరాడుతున్నాయి	నల్ల కుక్క మచ్చల కుక్క పోరాడుతున్నాయి
ఒక నల్ల కుక్క మరియు ఒక లైట్-రంగు కుక్క రోడ్డు మీద ఒకరినొకరు ఆడటం.	నల్ల కుక్క లైట్-రంగు కుక్క రోడ్డు మీద ఒకరినొకరు ఆడటం.
ఒక నల్ల కుక్క మరియు గోధుమ మచ్చలతో ఉన్న తెల్లని కుక్క వీధిలో ఒకరికొకరు చూస్తారు.	నల్ల కుక్క గోధుమ మచ్చలతో ఉన్న తెల్లని కుక్క వీధిలో ఒకరికొకరు చూస్తారు.
రోడ్డు మీద ఒకరినొకరు చూస్తున్న వివిధ జాతుల రెండు కుక్కలు.	రోడ్డు మీద ఒకరినొకరు చూస్తున్న వివిధ జాతుల రెండు కుక్కలు.
పేవ్మెంట్ మీద రెండు కుక్కలు ప్రతి ఇతర వైపు కదులుతాయి.	పేవ్మెంట్ మీద రెండు కుక్కలు ప్రతి ఇతర వైపు కదులుతాయి.

Figure 4.3 After removing block words

4.4 Word Embedding Generation

This fragment covers the utilization of fastText vector over the other monolingual and multilingual embedding such as IndicBERT, IndicFT, XLM-R and MuRIL. After we've cleaned and pre-processed the content, we'll go on to this step.

4.4.1 FastText

For text classification, we use fastText Telugu pre-trained monolingual embedding. These word vectors have been trained on Common Crawl and Wikipedia using the following hyperparameters:

- Model: CBOW model was used with position-weights
- Window size: 5-character n-grams, a 5x5 window, and 10 negatives
- Optimizer: stochastic gradient descent with respect to the negative log-likelihood
- Word vector dimension: There have been 300 uses of word vector dimensions
- Vocabulary size: 99882

4.5 Encoder Model Analysis

The comparative study of various pre trained CNN models is done. In our analysis, we understood that Inception-v3 performs better when compared to Resnet-50. The characteristics of all the analyzed models are mentioned.

4.5.1 InceptionV3

We are using InceptionV3 Telugu pre-trained model. The following are some architectural features to consider.

- Depth: 159
- Input parameters: 23,85,1784
- Additional Layer: Custom Softmax Layer
- Input Image scaling: Unscaled
- Drop Out: 0.5

```
class EncoderCNN(nn.Module):
    def __init__(self, embed_size, train_CNN=False):
        super(EncoderCNN, self).__init__()
        self.train_CNN = train_CNN
        self.inception = models.inception_v3(pretrained=True, aux_logits=False)
        self.inception.fc = nn.Linear(self.inception.fc.in_features, embed_size)
        self.relu = nn.ReLU()
        self.times = []
        self.dropout = nn.Dropout(0.5)

    def forward(self, images):
        features = self.inception(images)
        return self.dropout(self.relu(features))
```

Figure 4.4 Code

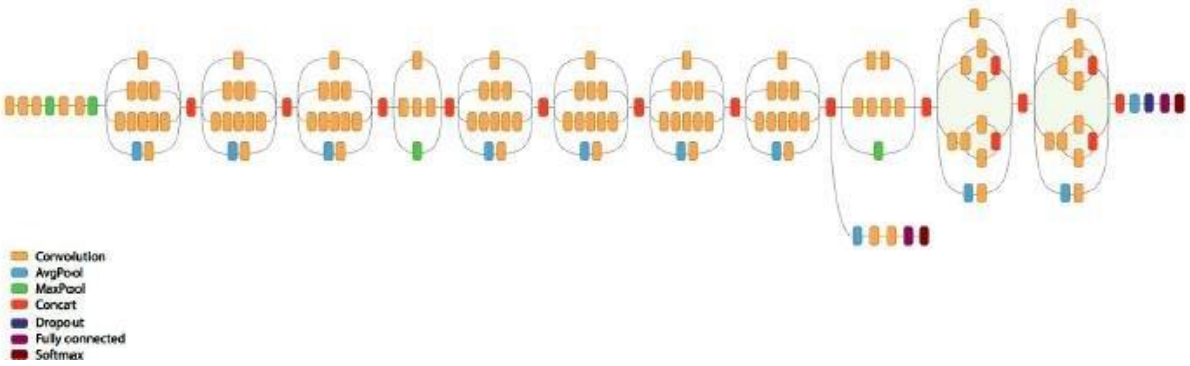

Figure 4.5 InceptionV3 Architecture

4.5.2 Resnet-50

The photos are passed through the resnet architecture in this section. The softmax layer at the top of the resnet layer is deleted. There are six trainable layers and 170 layers that cannot be trained. The output of the previous Crevice layer is 2048, and since the representation from this technique is tall, it may be passed through a thick layer of 256 neurons to equalize the representation from the 300-neuron artificial neural network (ANN). At this level, the layers are concatenated to produce 556 neurons, which are then transmitted through a custom softmax layer. The used enlargements are flat flip and pivot enlargements.

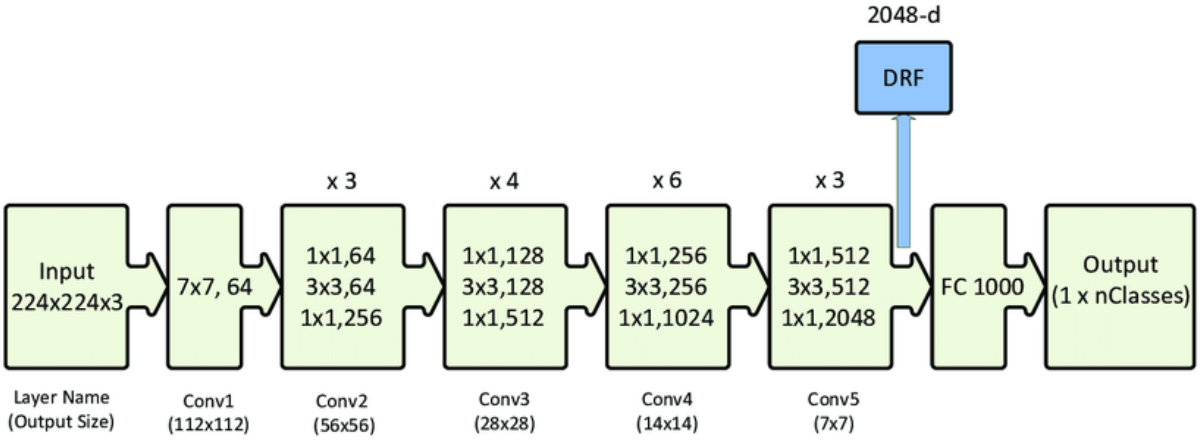

Figure 4.6 Resnet50 Architecture

4.6 Decoder Model Analysis

After performing tests with GRU and LSTM, we found LSTMs are better in terms of efficiency and speed. So, we use LSTMs in this research.

4.6.1 LSTM

The LSTM network has been imported from the keras.layers class of the Keras library. Despite the fact that LSTM offers a variety of parameter options, we evaluated a specific number of parameters for this study. The LSTM and dense layers have been employed for classification purposes.

LSTM hyperparameters:

- Hidden units: 128
- Dropout: 0.5
- Return_sequences: True

After applying various combinations of the different hyperparameters, we have moved forward with the above set of hyperparameters.

For the dense layer, the hyperparameters are:

- Units: 3
- Activation: Softmax

```
class DecoderRNN(nn.Module):
    def __init__(self, embed_size, hidden_size, vocab_size, num_layers):
        super(DecoderRNN, self).__init__()
        self.embed = nn.Embedding(vocab_size, embed_size)
        self.lstm = nn.LSTM(embed_size, hidden_size, num_layers)
        self.linear = nn.Linear(hidden_size, vocab_size)
        self.dropout = nn.Dropout(0.5)

    def forward(self, features, captions):
        embeddings = self.dropout(self.embed(captions))
        embeddings = torch.cat((features.unsqueeze(0), embeddings), dim=0)
        hiddens, _ = self.lstm(embeddings)
        outputs = self.linear(hiddens)
        return outputs
```

Figure 4.7 Decoder model

4.7 Summary

In this chapter, we reported the results and observations we obtained utilizing the research methodologies indicated in the preceding chapter. In the first section of this chapter, the data source and type, as well as the columns and their descriptions, are covered. Next, the exploratory data analysis and TTR of the given dataset were covered, validating the appropriateness of the text classification dataset. We also discussed the data cleansing procedures required to convert the raw data into the desired format.

Then we moved to encoder model analysis where we understood various types of pre-trained models are used and studied their respective efficiencies.

Finally, we analyzed the decoder mechanism and understood various embedding strategies like fastText. We analyzed the model with respect to various hyperparameters to obtain a state-of-art model.

CHAPTER 5

RESULTS AND DISCUSSION

This study is mainly to get a motive on producing captions of images on Telugu Language using encoder-decoder model. By using cleaned dataset and per image works remarkably which generates captions of high-quality after a comparison of four models, the model has

been trained. An account of smaller number of reference models to tone with, the BLEU score of the model is low. The results obtained from the research is very important as no study has been done to generate caption of images in Telugu. This project might be helpful for further study on image captioning in Telugu. The foremost method to calculate image captions in English called Human Evaluation and this seems to be true as evident from the result obtained from this project.

Thus, the project acquired a state-of-art result to generate image description in Telugu language. For training the image captioning model, this study uses machine-translated sentence.

This is no guarantee that the translation will be done perfectly with one hundred percent accuracy. Due to various grammatical structures of various languages, there will be a lot of complications for translating sentences from one language to other languages. Considering the limitations of machine translator, the experiments have to be done. In previous researches, it has been shown that the captions are extracted from crowd sourcing agents which consisting of very natural captions. For example, if we took an input of a cat doing some activity instead of drinking milk.

The statement "A cat is drinking milk" is being generated. This caption is repeated as numerous images of a cat are shown. This is because the model utilizes a greedy technique to estimate the next likely word based on the picture feature and the preceding word. Rapid overfitting of data occurs, and terms that occur less often in the text have lower probabilities. The optimal method for overcoming this issue is to use an attention mechanism that employs the same encoder-decoder architecture that assigns more weights to important images of text. This study lacked the resources and time to conduct studies on the attention mechanism. Nonetheless, it would be intriguing to see how Telugu picture captioning functions following the application of the attention mechanism. However, after applying attention mechanism it will be interesting to see how Telugu image captioning works. The future scope of this project is there will be a chance to generate description for images in Telugu by using attention mechanism.

For all training purposes, google colab was used. 6000 step size and batch size of 128 images are used as a set for each and every epoch. 16 hours of training time is estimated for each

epoch. CPU processing leads to training time which is highly estimated. The present was modified to run the code on a GPU which has an estimation time of forty-five minutes on each epoch. As the colab's idle time is one hour we need to check every one hour to keep on training the model. The model was brought about fairly with good captions but some of the captions were not relevant.

By decreasing the training time and to get better results variable tuning was done on the model. After review from various studies, a decision was taken to reduce the step size. Batch size is calculated as total train dataset size is divided by number of images set per batch so as to train the model with all the images used once per epoch. To utilize the maximum memory available the batch size was charged to 200 which makes the training filter. Because of the lack of computation power and the dataset was also huge reducing the step size was done. The model was again trained with new 100 epochs. This new model also generates captions to that of previous model.

For improved results, 50 epochs are added by doubling the previous step size and 50 epochs are added with a step size of three times the previous one. Each epoch whose extension is 'h5' and whose name corresponds to the epoch is kept by the model version. Finally, 200 models are stored and the belt model is formed since the minimal loss after training 15 epochs and the minimized loss after 100 epochs are identical. Additional 100 epochs significantly improve its performance.

Evaluation Model	Train	Test
BLEU - 1	0.32567	0.29871
BLEU - 2	0.31245	0.21345
BLEU - 3	0.21234	0.15023
BLEU - 4	0.15627	0.1231
NIST	2.2435	1.5342

Table 5.1 Test Results

5.1 Evaluation

5.1.1 BLEU

There is a metric named Bilingual Evaluation Understudy for calculating the accuracy. It's the best method and most preferable metric used for evaluating translations in machines. Despite the fact that the model is not a translation model, the output captions are compared to the original human-written captions of every given picture. The BLEU evaluation to get similarity score of 0 to 1 can be done on two sentences. Running metric over n-grams we can obtain different results. By using 'n' consecutive two words from predicted sentence in original sentence, the possible sub sentences in a sentence made. The BLEU score decreases if the value increases because since the larger n value mean predicted sentences are same as the reference sentence. By calculating average of BLEU scores perfect results can be obtained by predicting caption for each image. Caption cannot be generated at once it takes word by word. And it results more than this for computing the total dataset which contains 8000 images. The measure was applied independently to 50 randomly chosen train and test pictures. The table shown above compares all BLEU score findings derived from our model. The model has been trained over the training data which gives more BLEU score. 1-gram BLEU score ranges between 0.45 and 0.55 whereas 0.36 was achieved by our model which gives a good satisfaction. By obtaining cleaning techniques for caption, we can get 0.5 which can be nearly taken as a very good model.

N-Gram precision

$$p_n = \frac{\sum_{n\text{-gram} \in hyp} count_{clip}(n\text{-gram})}{\sum_{n\text{-gram} \in hyp} count(n\text{-gram})}$$

Bounded above by highest count of n-gram in any reference sentence

brevity penalty

$$B = \begin{cases} e^{(1-|ref|/|hyp|)} & \text{if } |ref| > |hyp| \\ 1 & \text{otherwise} \end{cases}$$

Bleu score: brevity penalty, geometric mean of N-Gram precisions

$$\text{Bleu} = B \cdot \exp\left[\frac{1}{N}\sum_{n=1}^{N} p_n\right]$$

Figure 5.1 Calculation of BLEU score

5.1.2 NIST

Initially, the score metric that used but we used NIST score metric for better understanding. It is an altered BLEU metric that adds mass to n-gram. The n-gram uniqueness to particular image or a set of images can be identified by Rarity of n-gram. If the model identifies elements and corresponds to certain image which get a high count. NIST score metric ran on fifty random train and test images individually. Because no reliable sources for comprehending NIST metric have been identified, BLEU is often employed. The assessment has been completed, and the findings have been compiled in case any reader wishes to compare.

CHAPTER 6

CONCLUSIONS & RECOMMENDATIONS

6.1 Introduction

This chapter focuses on the assessment of the whole academic project and provides comments. In the first section, we assess the aims and research questions posed in Chapter 1 and provide answers by detailing the research process. In the next part, we assess the investigation's

findings, as well as the limitations and challenges encountered. In addition, we address the contribution and relevance of NLP-related work. Finally, the future scope and suggestions are discussed.

6.2 Discussion and Conclusion

The project has shown that the usage of forthright approach, there is a possibility for designing captions for images model especially for the Indian Languages. It acquired really fair enough in an Indian language. This project helps to predict caption which is word after word with input of an image.

Subsequently, after evaluation we made use of BLEU score metric which received quality as an average but this can prove that for better captioning models, language will not be the barricade. For achieving outcomes which are similar that of high-level image captioning model, we can make use with the slightest adjustment and cleaning the models. Also, the better word embeddings can get results in par with current state-of-the-art models which are directed for English Language. We also need a good availability of dataset constitutionally wrote in Telugu can improve the results.

6.3 Limitations

We experienced various problems while working on this thesis, and we did our best to overcome them given our restricted resources and time availability. The hardware requirement as we need to train a huge dataset to attain a better model is quite a challenge. The pre-trained embeddings that we use in this research, consume a lot of RAM and needs CUDA cores for faster training. We limited the set of hyperparameters and trained in small batches to overcome this issue.

The data availability in Telugu language is one of the challenges we have faced. As the data gets increased, the results will be better too.

6.4 Recommendations

Our tips for improving the classifier's performance are listed below. To boost performance, data augmentation is strongly suggested. Data augmentation is a well-known and widely used way for increasing data by using various data augmentation strategies. This can help to solve the problem of data limits. We abstained from employing data augmentation techniques due to time constraints and the nature of this study, but it is possible that it will improve in the future.

Aside from data augmentation approaches, we strongly advise you to investigate deep and complicated neural network architectures that comprise a combination of CNN, dense layer, LSTM, and batch normalization. One of the most recommended architectural changes for machine learning modelling would be to replace the LSTM with attention-based models. Because attention-based models have outperformed LSTM in many downstream tasks, investigating them might result in improved performance.

We also advocate combining the investigated word embeddings with other NLP tasks such as Named Entity Identification (NEI), POS tagging, and Question Answering System (Q&A). These pre-trained embeddings may then be used in unsupervised tasks like information retrieval and extraction where we have minimal data.

6.5 Summary

We have discussed various analyses for this research study. This chapter validates the previously specified research approach, and then we analyzed the limitations of this study, along with the limits encountered throughout the research's execution. We also spoke about the influence and contribution this endeavor can have on the community. Finally, the chapter finishes with future recommendations and the possibility for more research in this field.

REFERENCES

Aker, A. and Gaizauskas, R. (2010) “Generating image descriptions using dependency relational patterns,” in ACL.

Anderson, P. et al. (2018) “Bottom-up and top-down attention for image captioning and visual question answering,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.

Aneja, J., Deshpande, A. and Schwing, A. G. (2018) “Convolutional Image Captioning,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.

Ardila, A., Bernal, B. and Rosselli, M. (2015) “Language and visual perception associations: meta-analytic connectivity modeling of Brodmann area 37,” Behavioural neurology, 2015, p. 565871. doi: 10.1155/2015/565871.

Bahdanau, D., Cho, K. and Bengio, Y. (2014) “Neural machine translation by jointly learning to align and translate,” arXiv [cs.CL]. Available at: http://arxiv.org/abs/1409.0473.

Bai, S. and An, S. (2018) “A survey on automatic image caption generation,” Neurocomputing, 311, pp. 291–304. doi: 10.1016/j.neucom.2018.05.080.

Bernardi, R. et al. (2016) “Automatic description generation from images: A survey of models, datasets, and evaluation measures,” The journal of artificial intelligence research, 55, pp. 409–442. doi: 10.1613/jair.4900.

Chen, F. et al. (2017) “StructCap: Structured Semantic Embedding for Image Captioning,” in ACM Multimedia.

Chen, F. et al. (2018) “GroupCap: Group-based image captioning with structured relevance and diversity constraints,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.

Chen, L. et al. (2017) “SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Chen, S. and Zhao, Q. (2018) “Boosted attention: Leveraging human attention for image captioning,” in Computer Vision – ECCV 2018. Cham: Springer International Publishing, pp. 72–88.

Chen, X. et al. (2018) “Regularizing RNNs for caption generation by reconstructing the past with the present,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.

Chen, X. and Zitnick, C. L. (2015) “Mind’s eye: A recurrent visual representation for image caption generation,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Cornia, M. et al. (2017) “Paying more attention to saliency: Image captioning with saliency and context attention,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1706.08474.

Cornia, M. et al. (2020) MeshedMemory Transformer for Image Captioning.

Cornia, M. et al. (2021) Universal Captioner: Long-Tail Vision-and-Language Model Training through Content-Style Separation.

Cornia, M., Baraldi, L. and Cucchiara, R. (2019) “SMArT: Training shallow memory-aware transformers for robotic explainability,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1910.02974.

Dai, B., Ye, D. and Lin, D. (2018) “Rethinking the form of latent states in image captioning,” in Computer Vision – ECCV 2018. Cham: Springer International Publishing, pp. 294–310.

Devlin, J. et al. (2018) BERT: Pretraining of deep bidirectional transformers for language understanding.

Donahue, J. et al. (2015) “Long-term recurrent convolutional networks for visual recognition and description,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Fang, H. et al. (2015) “From captions to visual concepts and back,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Farhadi, A. et al. (2010) “Every picture tells a story: Generating sentences from images,” in Computer Vision – ECCV 2010. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15–29.

Fei, Z.-C. (2019) “Fast image caption generation with position alignment,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1912.06365.

Frome, A. et al. (2013) DeViSE: a deep visual-semantic embedding model.

Gan, Z. et al. (2017) “Semantic compositional networks for visual captioning,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Ge, H. et al. (2019) “Exploring overall contextual information for image captioning in human-like cognitive style,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE.

Gu, J. et al. (2017) “An empirical study of language CNN for image captioning,” in 2017 IEEE International Conference on Computer Vision (ICCV). IEEE.

Gu, J. et al. (2018) Stack-Captioning: Coarseto-Fine Learning for Image Captioning. AAAI.

Guo, L. et al. (2019) “Aligning linguistic words and visual semantic units for image captioning,” in Proceedings of the 27th ACM International Conference on Multimedia. New York, NY, USA: ACM.

Gupta, A., Verma, Y. and Jawahar, C. (2012) “Choosing linguistics over vision to describe images,” Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence, 26(1), pp. 606–612. doi: 10.1609/aaai.v26i1.8205.

He, K. et al. (2016) “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

He, S. et al. (2021) “Image Captioning Through Image Transformer,” in Computer Vision – ACCV 2020. Cham: Springer International Publishing, pp. 153–169.

Hochreiter, S. and Schmidhuber, J. (1997) “Long short-term memory,” Neural computation, 9(8), pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Hodosh, M., Young, P. and Hockenmaier, J. (2013) “Framing image description as a ranking task: Data, models and evaluation metrics,” The journal of artificial intelligence research, 47, pp. 853–899. doi: 10.1613/jair.3994.

Hossain, M. D. Z. et al. (2019) “A comprehensive survey of deep learning for image captioning,” ACM computing surveys, 51(6), pp. 1–36. doi: 10.1145/3295748.

Hu, X. et al. (2021) “Scaling up vision-language pre-training for image captioning,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2111.12233.

Huang, L., Wang, W., Xia, Y., et al. (2019) “Adaptively aligned image captioning via Adaptive Attention Time,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1909.09060.

Huang, L., Wang, W., Chen, J., et al. (2019) “Attention on Attention for Image Captioning,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE.

Ji, J. et al. (2021) “Improving Image Captioning by Leveraging Intra- and Interlayer Global Representation in Transformer Network,” in AAAI.

Jia, X. et al. (2015) “Guiding the long-short term memory model for image caption generation,” in 2015 IEEE International Conference on Computer Vision (ICCV). IEEE.

Jiang, H. et al. (2020) “In defense of grid features for visual question answering,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Jiang, W. et al. (2018) “Recurrent fusion network for image captioning,” in Computer Vision – ECCV 2018. Cham: Springer International Publishing, pp. 510–526.

Karpathy, A., Joulin, A. and Fei-Fei, L. (2014) “Deep fragment embeddings for bidirectional image sentence mapping,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1406.5679.

Ke, L. et al. (2019) “Reflective decoding network for image captioning,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE.

Kipf, T. N. and Welling, M. (2016) “Semi-supervised classification with graph convolutional networks,” arXiv [cs.LG]. Available at: http://arxiv.org/abs/1609.02907.

Kiros, R., Salakhutdinov, R. and Zemel, R. S. (2014) “Unifying visualsemantic embeddings with multimodal neural language models,” in NeurIPS Workshops.

Krishna, R. et al. (2017) “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” International journal of computer vision, 123(1), pp. 32–73. doi: 10.1007/s11263-016-0981-7.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012) "ImageNet classification with deep convolutional neural networks," Communications of the ACM, 60(6), pp. 84–90. doi: 10.1145/3065386.

Kulkarni, G. et al. (2013) "Babytalk: understanding and generating simple image descriptions," IEEE transactions on pattern analysis and machine intelligence, 35(12), pp. 2891–2903. doi: 10.1109/TPAMI.2012.162.

Kuznetsova, P. et al. (2014) "TreeTalk: Composition and compression of trees for image descriptions," Transactions of the Association for Computational Linguistics, 2, pp. 351–362. doi: 10.1162/tacl_a_00188.

Li, S. et al. (2011) Composing simple image descriptions using web-scale n-grams.

Li, X. et al. (2020) Oscar: Object-semantics aligned pretraining for vision-language tasks.

Liu, F. et al. (2019) "Aligning visual regions and textual concepts for semantic-grounded image representations," arXiv [cs.CL]. Available at: http://arxiv.org/abs/1905.06139.

Liu, F. et al. (2020) Prophet Attention: Predicting Attention with Future Attention.

Liu, W. et al. (2021) "CPTR: Full Transformer Network for Image Captioning," arXiv [cs.CV]. Available at: http://arxiv.org/abs/2101.10804.

Liu, X., Xu, Q. and Wang, N. (2019) "A survey on deep neural networkbased image captioning," The Visual Computer, 35, pp. 445–470.

Lu, J. et al. (2017) "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Lu, J. et al. (2019) Vilbert: Pretraining taskagnostic visiolinguistic representations for vision-and-language tasks.

Luo, Y. et al. (2021) “Dual-Level Collaborative Transformer for image captioning,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2101.06462.

Mao, J. et al. (2015) Deep Captioning with Multimodal Recurrent Neural Networks (mRNN),” in ICLR.

Mitchell, M. et al. (2012) “Gener- ´ ating image descriptions from computer vision detections,” in ACL.

Mokady, R., Hertz, A. and Bermano, A. H. (2021) “ClipCap: CLIP Prefix for Image Captioning,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2111.09734.

Ordonez, V., Kulkarni, G. and Berg, T. (2011) Im2text: Describing images using 1 million captioned photographs.

Pan, J.-Y. et al. (2005) “Automatic image captioning,” in 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763). IEEE.

Pan, Y. et al. (2020) “X-linear attention networks for image captioning,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Pedersoli, M. et al. (2017) “Areas of attention for image captioning,” in 2017 IEEE International Conference on Computer Vision (ICCV). IEEE.

Qin, Y. et al. (2019) “Look back and predict forward in image captioning,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Radford, A. et al. (2018) Improving language understanding by generative pre-training.

Radford, A. et al. (2021) “Learning transferable visual models from natural language supervision,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2103.00020.

Ramanishka, V. et al. (2017) “Top-down visual saliency guided by captions,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Ren, S. et al. (2017) “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE transactions on pattern analysis and machine intelligence, 39(6), pp. 1137–1149. doi: 10.1109/TPAMI.2016.2577031.

Rennie, S. J. et al. (2017) Selfcritical sequence training for image captioning.

Sharif, N. et al. (2020) “Vision to language: Methods, metrics and datasets,” in Learning and Analytics in Intelligent Systems. Cham: Springer International Publishing, pp. 9–62.

Sharma, H. et al. (2020) “Image captioning: A comprehensive survey,” in 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC). IEEE.

Shen, S. et al. (2021) How Much Can CLIP Benefit Visionand-Language Tasks?

Shi, Z. et al. (2020) “Improving image captioning with better use of captions,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2006.11807.

Simonyan, K. and Zisserman, A. (2014) “Very deep convolutional networks for large-scale image recognition,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1409.1556.

Sugano, Y. and Bulling, A. (2016) “Seeing with humans: Gaze-assisted neural image captioning,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/1608.05203.

Szegedy, C. et al. (2015) “Going deeper with convolutions,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Tan, H. and Bansal, M. (2019) “LXMERT: Learning cross-modality encoder representations from transformers,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Stroudsburg, PA, USA: Association for Computational Linguistics.

Touvron, H. et al. (2021) Training data-efficient image transformers & distilla- ´ tion through attention.

Vaswani, A. et al. (2017) “Attention is all you need,” arXiv [cs.CL]. Available at: http://arxiv.org/abs/1706.03762.

Vinyals, O. et al. (2015) “Show and tell: A neural image caption generator,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Wang, L. et al. (2020) “Show, recall, and tell: Image captioning with recall mechanism,” Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence, 34(07), pp. 12176–12183. doi: 10.1609/aaai.v34i07.6898.

Wang, Y. et al. (2017) “Skeleton key: Image captioning by skeleton-attribute decomposition,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Wang, Z. et al. (2021) “SimVLM: Simple Visual Language Model pretraining with weak supervision,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2108.10904.

Wu, Q. et al. (2016) “What value do explicit High Level concepts have in vision to language problems?,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Xu, K. et al. (2015) “Show, attend and tell: Neural image caption generation with visual attention,” arXiv [cs.LG]. Available at: http://arxiv.org/abs/1502.03044.

Yang, X. et al. (2019) “Auto-encoding scene graphs for image captioning,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Yang, Y. et al. (2011) Corpus-guided ´ sentence generation of natural images.

Yang, Z. et al. (2016) “Review networks for caption generation,” arXiv [cs.LG]. Available at: http://arxiv.org/abs/1605.07912.

Yao, B. Z. et al. (2010) “I2T: Image parsing to text description,” Proceedings of the IEEE. Institute of Electrical and Electronics Engineers, 98(8), pp. 1485–1508. doi: 10.1109/jproc.2010.2050411.

Yao, T. et al. (2017) “Boosting image captioning with attributes,” in 2017 IEEE International Conference on Computer Vision (ICCV). IEEE.

Yao, T. et al. (2018) “Exploring visual relationship for image captioning,” in Computer Vision – ECCV 2018. Cham: Springer International Publishing, pp. 711–727.

Yao, T. et al. (2019) “Hierarchy parsing for image captioning,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE.

You, Q. et al. (2016) “Image captioning with semantic attention,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Zha, Z.-J. et al. (2022) “Context-Aware Visual Policy network for fine-grained image captioning,” IEEE transactions on pattern analysis and machine intelligence, 44(2), pp. 710–722. doi: 10.1109/TPAMI.2019.2909864.

Zhang, P. et al. (2021) VinVL: Revisiting visual representations in visionlanguage models.

Zhang, X. et al. (2021) “RSTNet: Captioning with adaptive attention on visual and non-visual words,” in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Zhou, L. et al. (2020) “Unified vision-Language Pre-training for image captioning and VQA,” Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence, 34(07), pp. 13041–13049. doi: 10.1609/aaai.v34i07.7005.

Zhu, X. et al. (2020) “AutoCaption: Image captioning with neural Architecture Search,” arXiv [cs.CV]. Available at: http://arxiv.org/abs/2012.09742.

J. Lu, J. Yang, D. Batra, and D. Parikh. (2018) “Neural Baby Talk,” in CVPR.

A. Karpathy and L. Fei-Fei. (2015) “Deep visual-semantic alignments for generating image descriptions,” in CVPR.