COMPARISON OF WHOLE SCENE IMAGE CAPTION MODELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TUGRUL GORGULU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2021

Approval of the thesis:

COMPARISON OF WHOLE SCENE IMAGE CAPTION MODELS

submitted by **TUGRUL GORGULU** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Ilkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Ilkay Ulusoy
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU _____

Prof. Dr. Ilkay Ulusoy
Electrical and Electronics Engineering, METU _____

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering, METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Assist. Prof. Dr. Erdem Akagündüz
Electrical and Electronics Engineering, Çankaya University _____

Date:10.02.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Tugrul Gorgulu

Signature :

ABSTRACT

COMPARISON OF WHOLE SCENE IMAGE CAPTION MODELS

Gorgulu, Tugrul
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ilkay Ulusoy

February 2021, 86 pages

Image captioning is one of the most challenging processes in deep learning area which automatically describes the content of an image by using words and grammar. In recent years, studies are published constantly to improve the quality of this task. However, a detailed comparison of all possible approaches has not been done yet and we cannot know comparative performances of the proposed solutions in the literature. Thus, this thesis aims to redress this problem by making a comparative analysis among six different models by implementing them. The selected models are generally trained only for the MsCOCO dataset in the literature. In order to make a more objective comparison, they are also trained for the Flickr30k dataset in this study. The selected models are as follows: Self-critical Sequence Training for Image Captioning [35], Neural Baby Talk [26], Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [3], Unsupervised Image Caption [12], Meshed Memory Transformer for Image Caption [7], and Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning [25]. First, the captions from all these models are extracted and the results are compared with the ones in their respective papers. In addition to popular metrics usually used in the papers, the captions from models are also evaluated by Word Mover's Distance and

BERT metrics. The findings of this thesis demonstrate that even though Bottom-up and Top-down attention and Neural Baby Talk can generate highly proper captions, Meshed Memory Transformer for Image Caption generally provides more promising results than the rest. Unsupervised Image Caption, on the other hand, is a far less successful algorithm since it does not use the direct relationship between images and their descriptions during the training stage.

Keywords: Survey, Comparison, Whole Scene Image Caption, Image Caption, Evaluation Metrics

ÖZ

FOTOĞRAFTAKİ BÜTÜN İMGELERİ ALTYAZILAYAN MODELLERİN İNCELENMESİ

Gorgulu, Tugrul
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Ilkay Ulusoy

Şubat 2021 , 86 sayfa

İmge altyazılama derin öğrenme alanının en zorlayıcı konularından biri olup amacı fotoğraf içeriğini dil bilgisi kuralarına dikkat ederek kelimeler ile ifade etme işidir. Bu alanda sürekli yayınlanan yeni makaleler metotların gelişmesini sağlamaktadır. Ancak modelleri inceleyen bir çok makale sadece yüzeysel ayrıntıları incelemektedir ve genellikle bilinen çalışmaları açıklamaktadır. Bu yüzden bu tezin amacı 6 farklı modeli programda çalıştırarak sonuçları yeniden üreterip algoritmaları ayrıntılı incelemektir. Bunun yanında seçilen modellerin çoğu sadece MsCoco datasetini kullanmaktadır. Daha sağlıklı bir karşılaştırma yapabilmek sonuçları olmayan bazı modeller Flickr30k datasetinde de eğitilip test edilmiştir. Seçilen modeller Self-critical Sequence Training for Image Captioning, Neural Baby Talk, Top-Down Attention for Image Captioning and Visual Question Answering, Unsupervised Image Caption, Meshed Memory Transformer for Image Caption ve Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning makalelerindeki modellerdir. Modeller çalıştırılarak altyazılar tekrar elde edilmiş olup metrik sonuçları makalelerdeki sonuçlar ile karşılaştırırmıştır. Ayrıca popüler metriklere ek olarak WMD ve

BERT gibi metrikler ile de üretilen cümlelerin değerlendirmesi yapılmıştır. Buna göre Neural baby talk ve Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering makaleleri başarılı sonuçlar elde etsede Meshed Memory Transformer for Image Caption makalesi en başarılı sonuçları göstermiştir. Bunun yanın da Unsupervised Image Caption en kötü sonucu göstermiştir.

Anahtar Kelimeler: Derleme, Karşılaştırma, Bütün Imgeleri Altyazılama, Imge Altyazılama, Değerlendirme Metrikleri

To the tree on Sartak sokak, Eskibağlar mah, Eskişehir

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my supervisor Prof. Dr. Ilkay Ulusoy for her guidance, support, and encouragement in the path of creation of this thesis.

I also would like to thank my family for their unconditional support during pursuing the master's degree. And, I want to thank my dog, Pablo, for his valuable friendship.

I give special thanks to Esat Kalfaoğlu for his help, guidance about academic life, and friendship. We have many good memories. Even though most of the time, our conversations do not have any point, I really enjoy them. Moreover, it was always a relief to know that I can always count on him.

I also want to thank Barış Can Çam, Barkın Evgin, Oğuzhan Babacan and Ayberk Aydın for their guidance about academic life and friendship. I might not be here without Barış Can Çam's support. Furthermore, I took most of the lectures with Barkın Evgin and Oğuzhan Babacan. It was a delight to study for exams and make homework with them. I won't forget the project that we make with Oğuzhan Babacan.

I also want to thank Berkay Yıldırım, Ahmet Çağrı Özcan and Şükrü Bakar for their friendship, all kinds of support and help. I won't forget good memories of Eskişehir and Ankara with them.

I also want to thank Faruk Çam and Recep Teke for their valuable friendship and conversations. I hope that we will spend more time together on Metu campus. I also want to thank Ebru Eryiğit for her support and encouragement.

I also would like to thank my co-workers in Tai for their supports.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

LSTM	Long-Short Term Memory
CNN	Convolutional Neural Network
NLP	Natural Language Processing
WMD	Word Mover's Distance
BERT	Bidirectional Encoder Representations from Transformers
CIDEr	Consensus-based Image Description Evaluation
METEOR	Metric for Evaluation of Translation with Explicit ORdering
BLEU	Bilingual Evaluation Understudy
SPICE	Semantic Propositional Image Caption Evaluation
ROUGE	Recall-Oriented Understudy for Gisting Evaluation

LIST OF SYMBOLS

a	image feature from CNN
v	embedded image feature
$\bar{m}$	embedded input of a LSTM
m	memory of a LSTM
i	input gate of a LSTM
f	forget gate of a LSTM
o	output gate of a LSTM
c	context vector
z	alignment score
α	attention coefficient
g	visual sentinel gate
x	input of a LSTM
h	hidden state of a LSTM
s	visual sentinel
β	gate for visual sentinel and context vector
p	word probability vector from the model
W	weights of fully connected vector
b	bias of fully connected vector
w	word
1_w	one-hot-code for a word
d	output of a LSTM
V	embedded image features set
j	region of a object on a image

y	ground truth word
q	output of a discriminator
r	reward
κ	confidence score for object's region
L	loss
γ	decay factor of a reward
λ	coefficient of a term
U	value vector of scale dot attention
K	key vector of scale dot attention
Q	query vector of scale dot attention
D_k	dimension size of a key vector
S	mask of decoder
Z	alignment score of scale dot attention
M	meshed memory
X	input of a transformer
Y	output of a transformer
ι	precision-recall weight
μ	a caption
ζ	a reference sentence
I	word dictionary

CHAPTER 1

INTRODUCTION

1.1 Introduction

Whole scene image caption task is one of the challenging problems in artificial intelligence. The aim is to transform the content of an image into one sentence. Since images and sentences have different structures and features, they are examined in two different fields: computer vision and natural language processing (NLP). Computer vision deals with image pixels to make them easier for machines to process. There are many computer vision applications in our daily lives, such as face recognition, optical character recognition, human tracking, etc. Natural language processing, on the other hand, examines languages to interpret words for machines. Application of NLP is also widespread in our lives, such as web searches, word recommendation during texting, language translation, etc. By merging these two fields, the whole scene image caption can understand the correlation of objects in the image and words in a sentence.

After recent developments in Artificial Intelligence's deep learning area, the whole scene image caption has become one of the hot topics. New methods are proposed every day. Furthermore, to improve new algorithms, current methods should be understood clearly. However, recent surveys do not focus on the details of the state of the art models. Moreover, they usually examine well-known studies. Thus, the motivation of this thesis is to analyze the improvements and weaknesses of different approaches in the whole scene image caption problem. In addition to that, re-generating captions from models can give us a chance to understand the flaws of the models. Another issue is that many papers in the literature utilize only one data set. A data

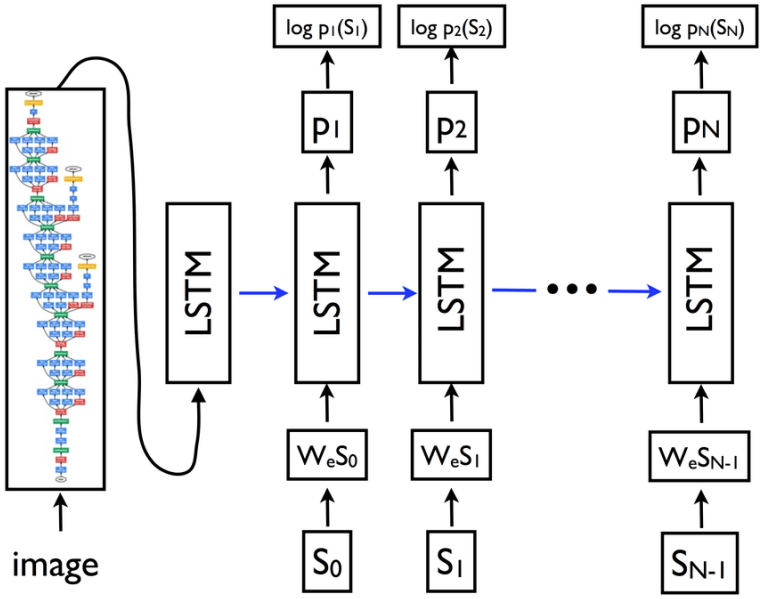

Figure 1.1: Show and Tell model

set for the image caption task should contain image-description pairs. Moreover, a description should give information about the corresponding image. However, these descriptions are not very clear to produce, such as labels in object detection or activity recognition. A description of an image might change even from person to person. Therefore, one data set might not be enough to compare the models.

Many excellent papers help to solve the whole scene image caption problem. On the other hand, the most famous article is "Show and tell model" [38] without any doubt. It is one of the first models which utilizes deep learning algorithms. The intuition comes from machine translation task. In the machine translation, a sentence in one language is sent to the encoder module. The encoder's output is taken by the decoder module and transformed into a sentence in another language. Therefore, rather than sending a sentence to the encoder module, an image is sent to the encoder in [38]. While encoder is defined as CNN (See Appendix A.1), LSTM (See Appendix A.3) is accepted as the decoder. In Figure 1.1, the details of the architecture can be seen. After [38], instead of using the image feature at the beginning of the decoder, [41] determines what activation grids the model should focus on using an attention map and presents it to the decoder at every step. Therefore, the model can look at different objects.

After improvements of [38] and [41], many papers are published for the whole scene

image caption problem. [36] proposes a new network called transformer which tries to take LSTM's place. Thus, some papers combine LSTM with a transformer to solve image caption problem, such as [15] and [29]. However, one of the first successful papers which reaches state-of-the-art performance using only the transformer is Meshed-Memory Transformer for Image Captioning [7]. Besides, there are also some improvements in image caption with unsupervised learning, such as [12], [22]. However, these models are still needed to be developed.

To improve a new algorithm, on the other hand, current models should be explained in detail. The useful information in the previous models might be used by the new methods. Furthermore, there are many valuable papers in the literature that help to solve the whole scene image caption problem. However, this thesis examines 6 methods, which have different approaches to the problem: Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning [25], Self-critical Sequence Training for Image Captioning [35], Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [3], Neural Baby Talk [26], Meshed-Memory Transformer for Image Captioning [7] and Unsupervised Image Caption [12]. In addition to that, drawbacks of the models are also explained for further developments.

Moreover, the results of the papers are obtained by implementing the code given in the references. In addition to metrics on the paper, captions of the models are evaluated by different metrics, such as Word Mover's Distance (WMD) and BERT in this thesis. For image-sentence pairs, two data sets are utilized: MsCoco and Flickr30k. In the literature, MsCoco is very popular because of the large number of samples. However, using one data set might not be enough to interpret the performance of the models because data sets might have some flaws. Therefore, to be sure about the performance of the algorithms, Flickr30k data set is also used, which was not applied in the original papers of Self-critical Sequence Training for Image Captioning [35], Meshed-Memory Transformer for Image Captioning [7] and, Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [3]. Therefore, the other contribution of this thesis is to show metric results for the Flickr30k data set. The benefit of using two data sets are revealed while comparing Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

[25] and Self-critical Sequence Training for Image Captioning [35]. Furthermore, the number of different captions table (See Table 4.5) might give us an idea of the novelty of generating captions.

CHAPTER 2

LITERATURE

2.1 Literature

Many models for image caption tasks have been designed after deep learning became popular. The standard algorithms used CNN (See Appendix A.1) as an encoder and LSTM (See Appendix A.3) as a decoder. Even though architectures of image caption task are continuously changed, CNN is, without any doubt, the best technique to extract spatial information from an image. Moreover, LSTM provides promising results for processing sequential data. Because of the memory cell, the network can produce a current sample based on previous samples.

The combination of the usage of encoder and decoder is, on the other hand, quite various in the literature. It can roughly be divided into six categories. Firstly, in Multi-modal networks, two encoders are used to extract both image and text features. After that, one decoder is utilized to produce captions. Secondly, Semantic networks focus not only on the classification of the objects on an image but also on the attributes of the objects. Thirdly, Attention models do only not use image features from CNN's last layer for once. While generating words, the models also learn where and what to focus on an image features. Fourth, the number of different objects and words in descriptions in datasets is minimal compared to the real world. To avoid this limitation, Zero-shot learning is used. Finally, typical image caption methods are based on supervised learning. Reinforcement learning and Unsupervised learning have also become quite popular recently. In the following sections, These six categories will be explained with related papers.

2.1.1 Multi-modal networks

Multi-modal networks are one of the first networks which utilize deep learning algorithms. As mentioned above, image features map and text features are used in the image caption. Moreover, a learning model should understand the relationship between image and text features. However, these features are not defined in the same space. Therefore, Multi-modal networks mapped them into the same space. One of the first paper which uses deep learning is [19]. In [19], corresponding descriptions of an image are examined in a parallel manner. In contrast to the previous model, it uses a convolution neural network to extract image features from the image. Moreover, it also utilizes a feed-forward neural network to map words in the datasets for word embedding. There are two different models in the paper. In the first model, Multimodal Log-Bilinear Models are used for mapping words and image features to the same space. While generating the current word, the summation of weighted embedding words and weighted embedded image features are sent to a softmax layer. It is also the first time that backpropagation helps the network to converge weight to their ideal values. The second method uses Factored 3-way Log-Bilinear, which examines image features like a gate to map image features to words. Then, it maps the result to the softmax layer. However, Multimodal Log-Bilinear is not the only method to map text features. [20] also encodes image features using CNN, but in this time, it utilizes RNN for text features. Then, these image features and text features are mapped to multi-modal space using a fully-connected neural network. Moreover, both image and word embedded vectors are joined. Then, a new vector embedding is created in a multi-modal space. Additionally, [20] also presents a new decoder method which is called Structure-Content Neural Language Model. The decoder learns to generate grammarly correct words using a given content embedding vector, word-specific structure, and previous words in this method.

As mentioned above, image features are used to generate a sentence. However, text features can also be used to reconstruct image features. For instance, [6] focuses on a bi-directional mapping between a sentence and an image. The idea comes from the illustration of an image by reading a sentence. However, to make this process easier, they try to re-generate image features instead of the whole image. During the train-

ing, the model attempts to reconstruct given image features and produce sentences simultaneously. Latent variables become a bridge between these two tasks. While re-generating image features, previous words are encoded to latent variables using RNN. Then, latent variables are transformed into image features. Besides, to generate sentences, image features are encoded to latent variables. Then, the current word is predicted based on previous words and latent variables.

Finding a proper sentence does not always need to generate a sentence. A similarity score can also be used. In [17], instead of producing a novel description, it ranks image-sentence pair and brings to the fore the sentence with the highest score. To find the most relevant image-sentence pair, all image-sentence pairs are assigned a score which is the inner product of visual embedding and text embedding. Besides, image features are obtained using RCNN, and fragments of visual embeddings and text embedding are converted to the multi-modal space using a neural network. Thus, the model is trained for both fragments of visual and text pairs and all fragments and sentences. In contrast to [17], some papers aim to generate more novel sentences. For example, J.Mao designed a new architecture which is called m-RNN in [28]. In this network, two embedding layers can convert the one-hot-encoded word vector to text embedding. Then, text features are presented to RNN. In addition to text features, image features are also presented to the RNN to avoid forgetting image representation. Then, text features of previous words and image features are mapped to a Multi-modal space using a neural network. The softmax layer is also applied to predict the current word.

2.1.2 Semantic Networks

Attributes are essential in the machine learning area. The color, shape, texture of an object are different types of attributes. CNN extracts only an abstract representation of a given image. Thus, even though an attribute's information is obtained at CNN's low-level layer, it might be lost at CNN's high-level layer. Semantic networks also aim to exploit all details of objects on an image using their attributes.

Attributes can be learned using other networks that focus on attributes only. For instance,[43] train additional an attributes detector using Multiple Instance Learning

[11], which is a weakly-supervised method, to detect attributes. As a decoder, there are three types of LSTM usage in [43]. To present these attributes to the network, they design LSTM-A. Moreover, five kinds of LSTM-A are mentioned in the paper. The differences are the order of presenting image features and attributes to the network. Also,[44] generates its captions with more than one attributes detectors using a top-down approach and a bottom-up approach. The difference between these approaches is explained as following: While the top-down method uses visual image representation and then starts producing words, the bottom-up process first generates words which describe image then merge them. [44] also combines these two approaches by calculating semantic attention. Besides, after image features are obtained, they are presented to the network only once. Then, detected attributes are used to compute the input attention score and output attention score. These two attention scores are utilized to create a new input vector to RNN and a new output vector for the Softmax layer.

Some papers ,on the other hand, inspires from the human brain to create more novel sentences. [39], for instance, claims that typical LSTM cannot really create novel captions from descriptions of images. The algorithms in the literature usually detect attributes before detecting objects in the image. Similar to human brain, [8] detects the objects first and then attributes. For this purpose, to train network for attributes and rest of the sentences, attributes and rest of the sentences are distinguished using Stanford constituency parser which classifies words in sentences as a noun, noun phrase, etc. In addition, there are two types of LSTM; Skel-LSTM and Attr-LSTM. Image features are first sent to Skel-LSTM. The skeleton of the caption is built. Then, Attr-LSTM is trained to give attributes for skeleton sentences.

In contrast to [39], [13] trains a new network to obtain attributes. While training, K common words in the description are detected for a given image, and they are considered semantic labels. Then, these semantic concepts are trained like multi-label classification. Afterward, detected semantic features are combined with word embeddings vectors and a hidden layer of LSTM for each step while generating output words for a given image.

2.1.3 Reinforcement Networks

Traditional image caption tasks use supervised learning. However, after recent improvements in Reinforcement learning, new papers have also started to be used Reinforcement algorithms in image caption tasks. In this learning mechanism, the model takes actions in an environment. Based on these actions, the model weights are updated according to reward value instead of a loss, such as in supervised learning. The purpose is to accumulate rewards as much as possible. In the Reinforcement learning aspect of image caption, the images are our environment and generated words from the model are the actions. Many models train a value network instead of using a reward function.

[34] is one of the models which utilizes a value network. In these tasks, there are two networks which are a policy network and a value network. Like supervised methods, the Policy network recommends a word to the model for each step using LSTM. After that, the value network tries to find the word with the highest score in the long-term. To do that, it concatenates embedding image features and text features and presents them to multi-layer perceptron to obtain a reward value. The policy network then updates their weights based on this reward. Another example is [45] on an actor-critic model for image caption tasks. The Policy network (Actor) produces a new word at each step while the Value network (critic) criticizes the current generated word. Rather than using MLP, [35] uses the Cider score function as a reward function. However, Cider cannot give a strong evaluation score by comparing only a word and a sentence. Thus, the Value network assumes that the Policy network would keep generating words based on current and previous words. The result caption is compared with descriptions. This process is also defined as an estimated future reward score for the current word. Then, Similar to [34], the Policy network is trained for the current word using this future reward score.

[8] aims to generate more natural, various, and semantically close to the ground truth. In [12], they claim that since traditional methods train their model based on maximizing likelihood estimation, these methods cannot create distinct captions from descriptions in datasets. Thus, [8] combines GAN and Reinforcement Learning to avoid this problem. It trains two different networks; Generator and Discriminator. Genera-

tor deals with producing sentences based on a given image. For diversity, a random vector is also expected as input. Words are stated as action and are decided based on the policy of the network. However, a bunch of words cannot be used to calculate a reward. For that reason, the Expected future reward is calculated for each word by letting generate the rest of the sentence at n times for the current word. After that, the average reward of these sentences is calculated, and Generator's parameters are updated. Besides, the discriminator is used as a reward function. It determines that a given sentence is from Generator or a sentence from the dataset. In the end, Generator and Discriminator are trained based on different objectives. While the Generator tries to deceive Discriminator about the generated sentence created by a human, Discriminator tries to understand the difference between a caption and a description in the dataset.

2.1.4 Unsupervised Networks

Unsupervised learning is another learning mechanism. In contrast to Reinforcement learning and supervised learning, the purpose is to teach a model to generate a proper caption without using any image-description pair. [22] is one of the first algorithms. The model transforms both texts and images to joint embedded space using RNN and CNN as an encoder. After extracting image features using CNN, image features are mapped to joint space using a multi-layer perception layer. To train MLP, a discriminator learns to distinguish joint embedded image features and visual concepts. [22] also aims to create connections between objects in the image and related objects in the image, such as 'a TV' and 'a TV table'. In this way, the model generates more novel sentences than the description in the dataset. Additionally, the RNN encoder learns to encode given text. In producing captions, the RNN decoder is very similar to the RNN encoder. However, in the former, text embedded vectors and image embedded vectors are received as inputs.

2.1.5 Zero-Shot Networks

The Zero-shot approach is needed to handle new objects which are not seen in the training step. These algorithms are essential, especially for wild image caption tasks. [40] also aims to utilize the objects in the datasets and a new object that CNN classified. Thus, at first, a caption is generated for a query image, but related objects in the sentence are not defined. Then, objects on the images are classified using a pre-trained Inception-ResNeV2 model, and objects' labels are put into the sentence. However, instead of presenting a noun to the network, <PL> token is given. Thus, LSTM is not affected while producing a caption when it encounters a new word. Moreover, there are no embedding vectors for nouns. While generating words, when LSTM encounters a noun, the corresponding hidden layer is transformed into visual representation space. To find a proper word, Write and Read Operation is defined in the paper: Write operation creates a visual representation-word pair during training while Read operation tries to find the closest visual representation-word pair for query visual representation.

Similar to [40],[14] offers to use words that are not contained in the training description. Similar to [22], unpaired image-description is used to train LSTM. However, paired image-description is also utilized in further steps. In [14], firstly, pre-trained CNN with the ILSVRC-2012 dataset is trained to extract semantic information from a given image. Besides, the most common verb, nouns, etc., are detected from training sentences, and they are used as labels while training CNN. Also, unpaired text corpora are utilized to teach LSTM the language model and teach the embedding layer a new word. Moreover, CNN and LSTM are combined by using the linear multi-modal unit. Outputs of both networks are converted to multi-modal space with linear layers. The output of the Multi-modal Unit is the currently estimated word. It is thus sent to the embedding layer before being sent to LSTM as a previous word.

2.1.6 Attention Networks

CNN provides only abstract information of an image. However, if LSTM takes this abstract information only at the first step, then the information might be lost in the last

step. Therefore, this information is better to give LSTM at every step. Moreover, the relationship between objects and words might be changed based on objects. It means that the model does not need to focus on the same object feature in every step. Thus, rather than using information the whole scene, attention models lead the networks to where and what the network should focus on image features during the production of words.

One of the first attention models is proposed in [16]. Instead of presenting a whole image feature, different objects on the image are extracted using segmentation and hierarchy boxes. In this way, rather than mapping the whole image features to one sentence, parts of the images are associated with the parts of the words. The current image features are calculated with a weighted sum of all found regions. The weights are computed based on the embedding vector of the previous word, the previous hidden state of LSTM, and the previous image features using a one-layer neural network. The model also focuses on the environment. Since different actions are done in different places, to generate more proper captions, the scene vector is also extracted. The model combines common words in descriptions and creates a topic vector. The image features are mapped to topic features using a multilayer perceptron. Moreover, the classified scene vector is presented to the input, forget, and output gate in LSTM for each step.

[41] model is proposed as an extension of the show, tell model. [41] extracts the image features using CNN. However, max pooling and fully connected layers are removed before sending image features to LSTM. Instead, the model trains multilayer perception to decide where and what to focus on for the next step. This attention mechanism is divided into two. Soft attention calculates a matrix whose size is equal to activation grids size on CNN. Every element in the matrix represents how strong the model focuses on that grid. The sum of the elements is equal to 1. On the other hand, hard attention gives only 1 to the grids in which the model looks. The other elements are set to zero. Thus, it only focuses on specific grids.

The attention models have usually examined only spatial area on the feature map. On the other hand, there are some papers that weight feature channel as well. For instance, [4] trains two different attention functions for spatial attention and Chan-

nel attention. In this manner, it tries to build more authentic relationships between image and description. Similar to [4], [42] also claims that conventional attention methods cannot have all the object information while generating words. Thus, to capture a more accurate relationship between objects, [42] trains a review network. After encoding both the raw image and corresponding description, a review network calculates a thought vector using hidden states of LSTM. Moreover, the review network is trained by using discriminative supervision. Besides, [42] can convert the classic attentive encoder-decoder model by replacing thought vectors and review vectors with hidden state and context vectors. However, the mentioned attentive model is very different than [41] since image features are obtained from a fully connected layer instead of a low convolution layer.

Instead of only modifying LSTM and CNN's activation grids, object proposal regions and spatial transformer can also be used to improve an attention mechanism. For example, the activation grids model in [32] is used for comparing with other region models. Furthermore, the object proposal region model finds edge-boxes of objects on grids. Finally, in the spatial transformer, an affine transformation is applied to every grid. Then, the features map is re-sampled. Moreover, in contrast to previous models, [32] creates a hidden layer, image region, and word embedding pairs. Thus, the probability of the next region and words is calculated by combining MLP for previous words, hidden state, and region.

[24] focuses on evaluating generated attention regions in [41] to understand the attention mechanism more clearly and reduce its drawback. Word-region maps in Flick30k Dataset are utilized to evaluate them with human perception. After the model generates the region map, the ground truth region is placed on the attention map. Besides, the sum of the common grids probability represents the score of generated attention. Besides, [24] also claims that [41] does not exploit the advantage of the prior probability of ground truth attention. Thus, it trains a new model that learns to produce an attention map based on the ground truth region and creates a relationship between region and word.

Attention mechanism can also be used to calculate long dependency. Scale-dot attention is very common in transformer networks (See Appendix A.4). The network uti-

lizes self-attention mechanism to understand the relationship between words. Therefore, in [15], the transformers are utilized to refine image features before sending them to the decoder and extracting a better relationship between image features and words. Besides, [15] calculates information and gate vector by merging LSTM and transformer. Multiplication of these two vectors provides a new context vector to the model.

CHAPTER 3

METHODS

This chapter aims to explain image caption methods with more details. Most of the models in this chapter were derived from highly cited papers "Show and Tell model" [38] and "Show, Attend and Tell" [41]. In [38], content of the image is extracted using CNN (See Appendix A.1). However, rather than training a new model, a CNN model, which was trained for object classification, is used after removing the last fully connected layer (See Appendix A.1) from the network. Then, the last feature map from CNN is converted to a vector by taking the average in every dimension of a feature map. Therefore, the output of CNN presents an abstract representation of an image. To exploit this abstract information, LSTM (See Appendix A.3) is used in [38]. Moreover, in every step, LSTM generates a new word based on abstract information and previous words.

Besides, instead of taking the average in every dimension of the feature map, [41] calculates coefficient for every grid of the feature map. Then, multiply these coefficients with corresponding grids. These values of the coefficient are updated in every step of LSTM. Thus, LSTM focuses on different places on the image while generating different words. This approach is called attention in the literature.

As mentioned in Chapter 1, this thesis examines six approaches to solve the image caption problem. These methods are chosen because they all have different perspectives to solve the whole scene image caption problem. The methods are as follows: Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning [25], Self-critical Sequence Training for Image Captioning [35], Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

[3], Neural Baby Talk [26], Unsupervised Image Captioning [12], Meshed-Memory Transformer for Image Captioning [7].

3.1 Image Caption Model Selection

Since this thesis aims to compare image caption models, academic papers with different perspectives were selected for comparison. The first method, Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning (See section 3.2), is the one closest to the traditional approaches such Show and Tell method [38] or Show, Attend and Tell method [41]. Yet, it brings a brand-new concept to the literature and considers not looking at image features while generating a non-visual word. It belongs attention categories in the Chapter 2. Besides, traditional methods are generally trained by using supervised learning. The second method, Self-critical Sequence Training for Image Captioning (See section 3.3), brings a successful reinforcement model to the image caption area by updating its weights by metrics rather than a loss function. Thus, it can be considered one of the model in Reinforcement categories. The third model, Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering (See section 3.4), tries to improve performance by changing the encoder module while many conventional image caption tasks focus on the decoder module. It also extracts image features of the object bounding boxes rather than whole image features. In addition, it might be categorized as attention network. The fourth model, Neural Baby Talk (See section 3.5), claims that many methods create similar sentences to the sentence in the training datasets. Therefore, it tries to generate novel sentences. Besides, it belongs to attention network categories in Chapter 2. The fifth model, Meshed Memory Transformer for Image Caption (See section 3.7) is currently one of the state-of-the-art methods in the Image Caption area. Moreover, in deep learning area, it is one of the first models which does not utilize LSTM for word prediction. Since model both exploits attention and reinforcement methods, it might be belong to these two categories. Finally, the sixth method, the Unsupervised Image Caption (See section 3.6), is the first unsupervised task in this area. The model can be utilized not only for the images in the prepared datasets but also for random images on the Internet. Moreover, it can be considered in Unsuper-

vised learning categories.

All in all, all these methods examine different parts of the image caption area, yet they bring new ideas to the literature.

3.2 Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

One of the selected methods is Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning [25]. In this section, details of the model will be explained. The adaptive attention was designed to reduce the drawbacks of [41]. One of the drawbacks of [41] is to focus on where and what to look at the image for every word. To accomplish this task, while producing each word, the network generates probability distribution for the feature map's activation grids. In other words, an attention map is calculated to focus on a different part of the image to generate a proper word. However, words are not equally related to the image. For instance, 'the' or 'of' words do not have direct visual appearances on the image. Thus, the network tries to generate non-visual words using image features. Besides, an attention map might not lead to a correct place on the image since the correct place does not exist while producing these non-visual words. Therefore, examining the image at every step might not be a practical approach. For this reason, adaptive attention [25] focuses on when to look at the image.

3.2.1 Algorithm

3.2.1.1 Spatial Attention

While looking at the image, the model still utilizes the spatial attention mechanism which tells the model where and what to look at the image at every step. On the other hand, in this method, image features are divided into two categories: global image

feature v_g and spatial image feature v_n.

$$\begin{aligned} v_n &= ReLU(W_a a_n)\ for\ n = 1\ to\ k \\ v_g &= ReLU(W_b a^g) \end{aligned} \tag{3.1}$$

Activation grids are defined as $A = a_1, a_2, ..., a_k\ a_n R^{2048}$. Global image feature v_g is calculated as average of all grids a_g.

$$\hat{m}_t = \tanh\left(W_c[h_{t-1}, x_t] + b_m\right) \tag{3.2}$$

$$m_t = f_t \cdot m_{t-1} + i_t \cdot \hat{m}_t \tag{3.3}$$

$$h_t = o_t \cdot \tanh\left(m_t\right) \tag{3.4}$$

In contrast to [41], the global image features are presented to the network at every iteration by using x_t which is equal to $[w_t; v_g]$ where w_t is word embedding vector. On the other hand, W_c is a fully connected layer. In addition, while m_t is memory cell and h_t is hidden state, f_t, o_t,i_t are LSTM's gate.

$$z_t = W_h^T \tanh\left(W_v V + (W_g h_t)1^T\right) \tag{3.5}$$

$$\alpha_t = softmax(z_t) \tag{3.6}$$

During the calculation of the α attention map, spatial image features are utilized. Therefore, V is equal to [v_n] for n=1 to k. Hidden State is defined as output for LSTM. Thus, the hidden state and image features are mapped to the same space using a fully connected layer. To add non-linearity, the tanh function is used. Formula 3.5 is also called additive alignment score in the literature. It aims to calculate the similarity score for two vectors. Then, the result is a pass to the softmax layer to normalize the result. Since $\sum_{n=1}^{k} \alpha_{tn} = 1$, this attention mechanism also can be defined as soft attention.

$$c_t = \sum_{n}^{k} \alpha_{tn} v_{tn} \tag{3.7}$$

Besides, a context vector is computed by multiplied image features with an attention map. After these operations, c_t acts like image features.

3.2.1.2 Adaptive attention

The network learns when to look at the image using adaptive attention. Instead of directly examining the spatial image features, the model focuses on global visual features and previous text features. Moreover, since x_t is equal to $[w_{t-1}; v_g]$, LSTM also knows global information about the image. Similar to the gates in LSTM, adaptive attention decides this 'when' action by calculating the gate.

$$g_t = \sigma(W_x x_t + W_h h_{t-1}) \tag{3.8}$$

$$s_t = g_t \cdot \tanh(m_t) \tag{3.9}$$

In literature, s_t is also called visual sentinel. To calculate it, weights are used to map both input and hidden layers to the same space. Additionally, the sigma function is utilized to make this map operation non-linear. g_t is calculated to determine how much information the network will use from LSTM memory. Since the sigma function's range is in [0,1], the information from memory might not be used if the gate value is zero.

$$\hat{c}_t = \beta_t s_t + (1 - \beta_t) c_t \tag{3.10}$$

What features will be examined is determined using the Beta gate. To calculate the beta gate, an α attention map is calculated as below. The last element of the attention map represents the beta gate value. W_h^T is a vector.

$$\hat{\alpha}_t = softmax([z_t; W_h^T \tanh(W_s s_t + (W_g h_t))]) \tag{3.11}$$

The current word's probability is calculated as follows:

$$p(w_t^d) = softmax(W_p(\hat{c}_t + h_t)) \tag{3.12}$$

3.2.2 Objective function

The cross-entropy loss function is used in the model. The general formula of cross-entropy as below

$$H_{p_m}(p_g) = -\sum_{n=1}^{N} p_g(w_n). \log(p_m(w_c)) \tag{3.13}$$

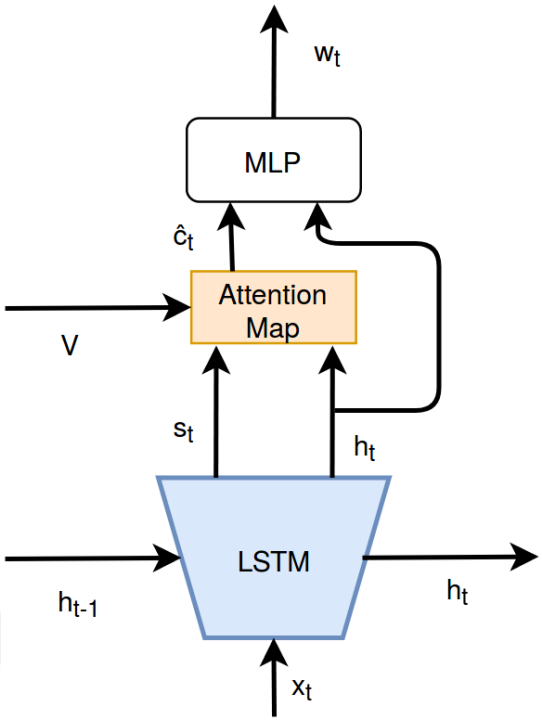

Figure 3.1: Adaptive Attention Architecture

In [25], $p_g(w_c)$ is equal to 1 since ground-truth word probability is always 1. Probability of predicted word from the model represents as $p_m(w_c)$. Then the equation becomes as below;

$$\log p_m(w) = \sum_{t=1}^{T} \log \left(p_m(w_t | w_1, ..., w_{t-1}, V) \right) \tag{3.14}$$

Besides, while generating a caption, teacher forcing is applied. Teacher forcing means that whatever the current word the model generates, the ground-truth word is presented to the network as the input for the next step. Therefore, $log(p_m(w_t|w_{t-1}, w_{t-2}, .., V))$ is defined as the probability of generated word for given ground-truth's words w_1,w_2,...,w_{t-1} and the image feature V. To obtain the minimum loss, Adam optimizer is used. Also, to compute the gradient value in Adam optimizer, Backpropagation is utilized.

3.3 Self-critical Sequence Training for Image Captioning

Another selected algorithm is Self-critical Sequence Training for Image Captioning [35]. The self-critical method focuses on some disadvantages of the traditional image caption task. For instance, as mentioned earlier above, adaptive attention [25] is

trained using Teacher Forcing methods. On the other hand, while testing the model, the current output is used as an input for the next step in LSTM. However, if a wrong word is produced, LSTM also has to produce the next words based on the wrong word. Since the model never experiences this event during the training process, the model probably produces inappropriate words. As the model keeps generating words, the sentence might be full of unrelated words to ground-truth captions. This problem is called exposure bias in the literature. Secondly, in the training steps, weights of the traditional image caption models are updated based on the cross-entropy loss function. However, in the testing step, metrics such as CIDEr and BLEU are utilized to evaluate the generated captions with the dataset's descriptions. Therefore, even though the loss might be very low during the training process, the metrics might still give a low score to captions. For these reasons, the Self-critical model [35] is one of the reinforcement learnings which aims to overcome both issues mentioned above.

Since metrics are non-differentiable, they cannot be used to calculate the gradient in the optimization step. However, metrics can be used as a reward function in Reinforcement learning. Therefore, the network directly optimizes its weights based on metrics. Moreover, instead of taking the average of the reward values of generated samples from the training step and using it as a baseline, the captions from the test step are utilized as a baseline in the REINFORCE algorithm. Therefore, Self-critical [35] also observes what is happening at test time during the training process. In that manner, the model also aims to reduce variance.

3.3.1 Algorithm

Even though Self-critical [35] exploits the advantages of reinforcement learning, in the first 30 epochs, it uses traditional supervised learning. Cross entropy is used as a loss function, and ground-truth words are presented to the network using Teacher Forcing.

$$x_t = W_e 1_{w_{t-1}} \; for \; t > 1, x_1 = V \tag{3.15}$$

$$i_t = \sigma(W_{ix} x_t + W_{ih} h_{t-1} + b_i) \tag{3.16}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{3.17}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{3.18}$$

$$m_t = i_t \cdot \phi(W^{\otimes}_{zx}x_t + W^{\otimes}_{zh}h_{t-1} + b^{\otimes}_z) + f_t \cdot m_{t-1} \tag{3.19}$$

$$h_t = o_t \cdot \tanh(m_t) \tag{3.20}$$

$$d_t = W_d h_t \tag{3.21}$$

$$p_t \approx softmax(d_t) \tag{3.22}$$

As can be seen, after the image features are extracted using CNN, they are sent to LSTM as the input vector. After the first step, the one-hot-encoded word is transformed into embedding space using W_e. Since it is the only input vector of the LSTM, it is defined as x_t in the formula. Similar to symbology of [25] , i_t , f_t , o_t are gates in LSTM while m_t is memory and h_t is hidden layer.

3.3.2 Image caption task in the Reinforcement learning aspect

As was explaining above, after 30 epochs, reinforcement learning starts to be used in the image caption. LSTM is used as an agent. Image and text features are defined as the environment. Based on the environment, the agent takes an action ,which is a word in our task, for each step. Since the action is chosen according to policy, LSTM's weights can be defined as our policy. Moreover, new action leads the agent to a new state which is a hidden state and memory cell. Also, every action should be either rewarded or punished to lead the model to a better state for future decisions. Thus, reward function (metrics) are utilized to evaluate our actions. The purpose of this reinforcement learning is to have a minimum negative expected reward at the end.

$$L(\theta) = -E_{w^d \approx p_\theta}[r(w^d)] \tag{3.23}$$

Adam optimizer is used for optimization. As a gradient, the below formula is utilized.

$$\nabla_\theta L(\theta) = \sum_{t=1}^{T} \frac{\partial L(\theta)}{\partial d_t} \frac{\partial d_t}{\partial \theta} \tag{3.24}$$

$$\frac{\partial L(\theta)}{\partial d_t} = (r(\mu^s) - r(\mu^a))(p_\theta(w_t|h_t) - 1_{w_t}{}^d) \tag{3.25}$$

While μ^s is sampled from the training step, μ^a is from the test step. Moreover, $p_\theta(w_t|h_t)$ is the probability distribution of words from the test step. However, $1_{w_t^d}$ one-hot code for the word is generated in the training step. Besides, even though words are generated from different steps, both steps utilize the same weights. $\frac{\partial d_t}{\partial \theta}$ is, on the other hand, equal to the difference between calculated word probabilities after updating weights. Besides, instead of using the maximum probability of word probabilities in training mode, the network generates random samples based on multinomial distributions to increase the long-term reward. In addition to that, to reduce the variance, the reward from test mode is used as baseline in REINFORCE with a baseline formula 3.26. (To make clear,test mode does not use any test image in training step. The model still uses training images but it shows test step's behaviour). According Self-critical model [35], sample number is 5 in Beam Search during generating captions for test's images.

$$\nabla L(\theta) = (r(\mu^s) - b)\nabla\, log(p_\theta(\mu^s)) \tag{3.26}$$

To give rewards to the current generated w_t, the network keeps generating the words until receiving the end token. After creating a full sentence, the captions can be sent to the reward function, and $r(\mu^s)$ is calculated for captions from the training step. Also, words for $r(\mu^a)$ are computed based on the argmax of softmax's output in the test step.

3.4 Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

Top-Down and Bottom-Up Attention [3] is another model this thesis examines. In many conventional image caption tasks in the literature, the models determine object

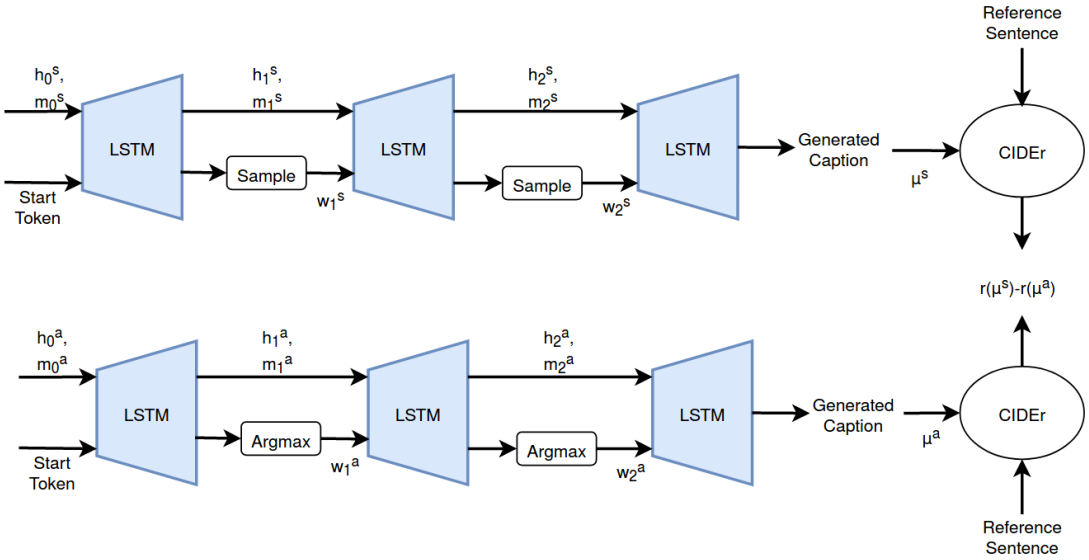

Figure 3.2: Self-critical Architecture

region based on equally separated activation grids. However, not every object is able to fit in the border of exact grids. Usually, many attention models decide what grids the model uses while generating words. Thus, Top-Down and Bottom-Up paper [3] focuses on extracting the salient object region before producing words. By doing so, the model also solves equal grids region problems, such as in Figure 3.3. Therefore, while Bottom-up attention aims to extract salient object regions, Top-down attention focus on generating word in a much more traditional perspective.

3.4.1 Algorithm

As mentioned above, instead of utilizing all activation grids of the last layer of CNN, the objects on the image are detected using Faster-RCNN. Object detection algorithms contain two steps. At first, regions of objects are detected using Region Proposal Network. The intuition is that every grid might be the midpoint of the object on the image. Therefore, the network utilizes anchor boxes to determine whether there is an object or not. A proposal region and ground-truth region are compared using Intersection over Union score. In this score, $score = \frac{AreaofOverlap}{AreaofUnion}$. The regions that have a higher overlap score than a threshold are used. Then, to the class region, the corresponding region map is sent to a fully connected layer. However, since the sizes of objects are different, the proposal region's size differs from image to image.

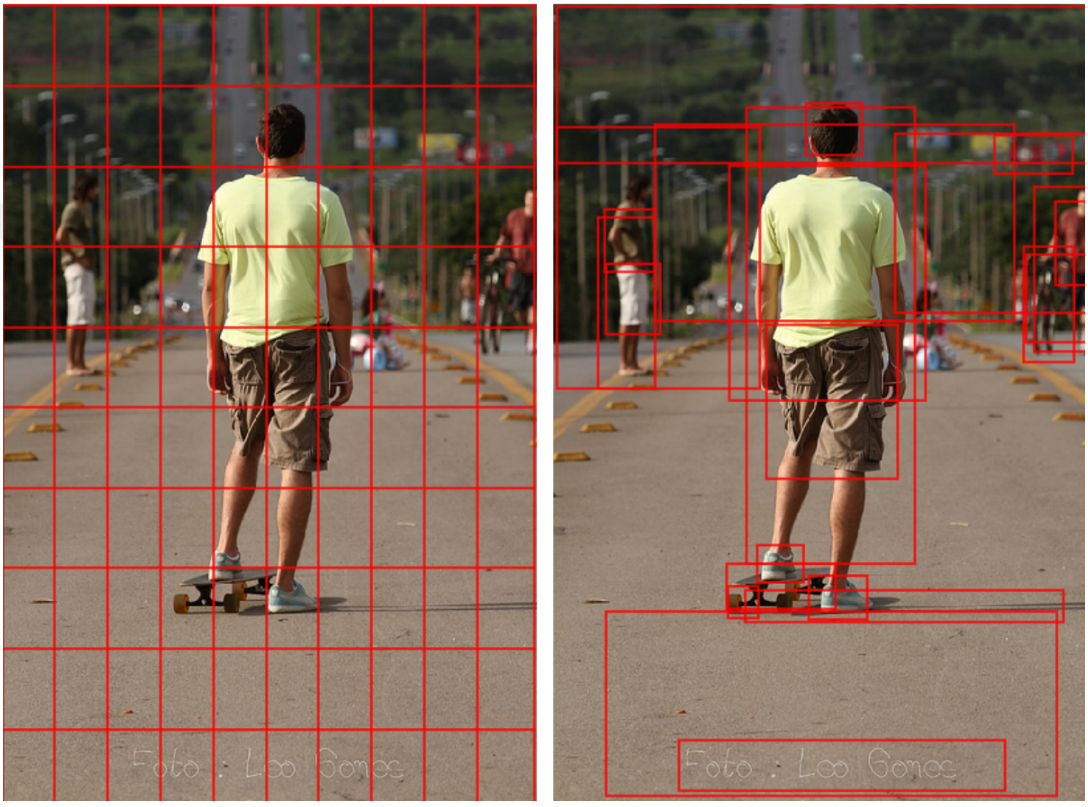

Figure 3.3: Left picture shows that the model examines only equal activation grids of CNN's last layer. Right picture shows that the model extracts object's feature using Faster-RCNN. The picture is taken from [3]

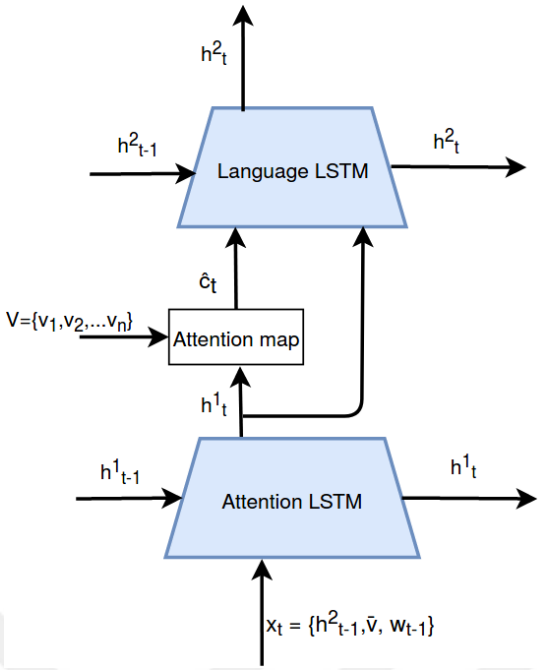

Figure 3.4: Top Down and Bottom Up attention Architecture

Hence, it brings difficulty for a constant size fully connected layer. RoI pooling is applied to solve this issue. In this method, the Region is divided by blocks. The size of the blocks is one of the parameters learned during training. Moreover, the mean of the values in the blocks is calculated and sent to the last layer. The channel number is not affected in this operation. Since a specific part of the image is used, and the rest is ignored, this attention mechanism is called Hard-attention. The outputs of Faster-RCNN are a class label and attributes if there are any.

In the Top-down attention part, two LSTM layers are divided as attention LSTM and Language LSTM. In attention LSTM, a hidden state is computed to calculate an attention map and to generate words. As an input, $x_t^1 = [h_{t-1}^2, \bar{v}, w_t]$ is used. $\bar{v}$ is the average sum of the regions v_n that is found from Bottom-Up attention. w_t is a word embedding vector for the corresponding word. Similar to the previous attention methods, the attention map is calculated as;

$$z_{n,t} = W_a^T \tanh\left(W_{va} v_n + W_{ha} h_t^1\right) \tag{3.27}$$

$$\alpha_t = softmax(z_t) \tag{3.28}$$

Since scores are normalized using the softmax function, the sum of the score is equal

to 1. In that manner, this attention mechanism refers to soft attention. Moreover, the sum of weighted image features are calculated as follows;

$$c_t = \sum_{n=1}^{K} \alpha_{n,t} v_n \tag{3.29}$$

For second LSTM, the input is $x_t^2 = [c_t, h_t^1]$ and the current word is found using $p(w_t|w1:t_{-1}) = softmax(W_p h_t^2 + b_p)$

3.4.2 Objective function

As an objective function, the cross-entropy loss function is used

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log p_\theta(w_t^*|w_{1:t-1}^*) \tag{3.30}$$

3.5 Neural Baby Talk

In this section, the algorithm of Neural Baby Talk [26] will be examined. In many image caption algorithms, the primary purpose is to generate a proper sentence for a given image. However, [26] claims that producing a caption cannot be beyond copying sentences in training datasets. For instance, if many pictures are related to a cat in front of the window picture in the training state, when the model sees an animal in front of the window in the test state, the animal is automatically defined as a cat. Thus, capturing new cases in the image is very limited in many methods.

For that reason, Neural Baby Talk [26] claims that it generates a template for textual words at first. It links visual words to regions on the image. Then, the objects on the regions are classified, and the class word is put in the corresponding place on the template. In that way, the model can create more novel sentences by classifying the objects using a different classifier.

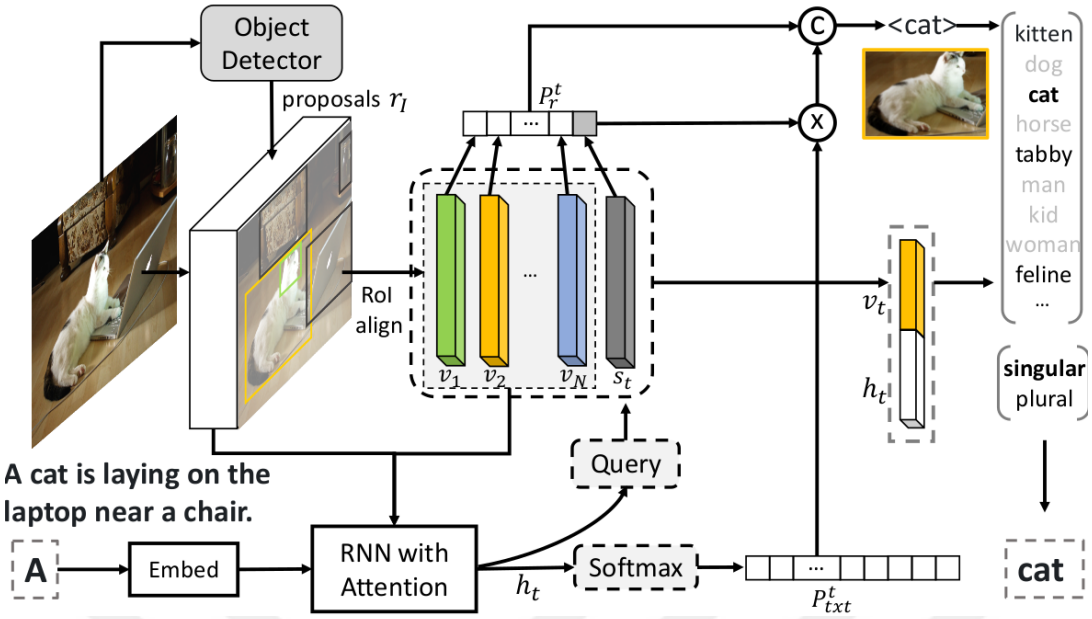

Figure 3.5: Neural Baby Talk Architecture [26]

3.5.1 Algorithm

Neural Baby Talk [26] generates its words using maximum likelihood estimation and cross-entropy loss function as the conventional image caption model.

$$p(w_t|w_{1:t-1}, V) = p(w_t|j_t, w_{1:t-1}, V)p(j_t|w_{1:t-1}, V) \tag{3.31}$$

In the above formula, while $w_{1:t-1}$ are words and V is embedded image feature, j_t is an object region on the image. While generating textual words , the visual sentinel is calculated as similar to adaptive attention (See section 3.2).

Furthermore, a pointer vector is computed to focus on a specific region of the image. v_t is extracted using Faster-RCNN for each region. The probability of which region is used is calculated as follows;

$$z_i^t = W_h^T \tanh\left(W_v v_t + W_z h_t\right) \tag{3.32}$$

$$P_{rI}^t = softmax(z^t) \tag{3.33}$$

$$P_r^t = softmax([z^t; W_h^T \tanh(W_s s_t + W_z h_t)]) \tag{3.34}$$

In the above formula, s_t might be defined as a visual sentinel feature of $\hat{j}_t$. While W_{th}^T $\tanh(W_s v_t + W_z h_t)$ represents $p(\hat{j}_t|w_{1:t-1}, V)$, probability of the generated current

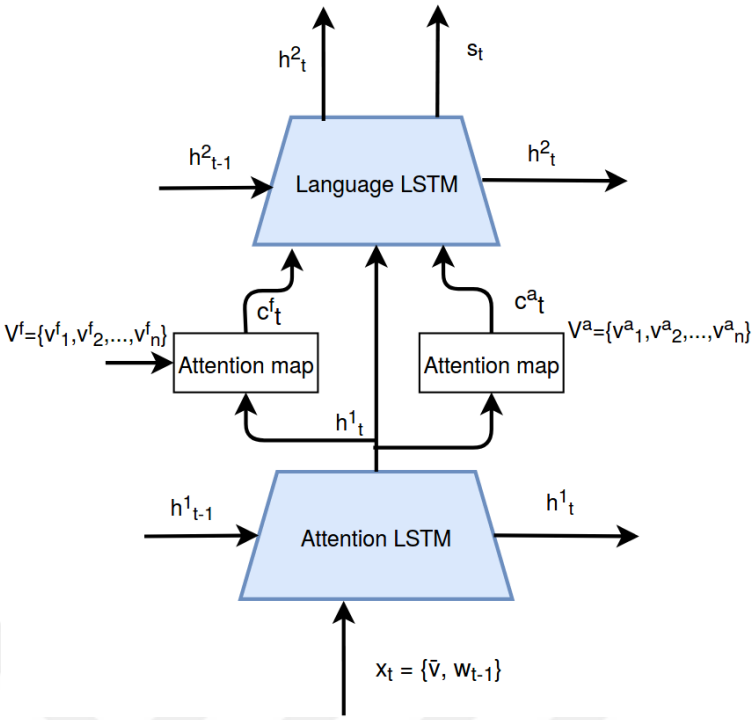

Figure 3.6: Language Architecture for Neural Baby Talk

word $P^t_{txt} = softmax(W_q h_t)$ is equal to $p(w_t|\hat{j}_t, w_{t-1}, V)$. However, before putting the class label, the object should be defined properly. Hence, whether the noun is singular or plural is determined by a fully connected layer. In addition, to create more novel sentences, fine-grained is defined for specific words. For instance, if <bird> is chosen as Object Category, the kind of the bird ,i.e whether it is an owl,seagull, duck etc., can be detected by the fine-grained network.

$$p_{bt} = softmax(W_b ReLU([v_T; h_t])) \tag{3.35}$$

$$p_{gt} = softmax(W_u^T W_g ReLU([v_T; h_t])) \tag{3.36}$$

Where p_{bt} is the probability of whether the noun is singular or plural, p_{gt} is the fine-grained probability which ReLU activation function. Besides, during creating fine-grained data for nouns, pre-trained glove word embedding is utilized. Moreover, W_u^T is used to map object category word to fine-grained class in the dataset.

Even though visual sentinel is produced as similar to Adaptive Attention [25] (See section 3.2), Attention model is different from Adaptive attention. There are two layers of LSTM in the model. The first one calculates hidden states to obtain attention

maps. While doing so, in this time, detected regions from Faster-RCNN and activation grids from CNN are used together. The second one uses these attention maps to generate the next word. In the end, α attention mechanism is calculated as follows;

$$z_t = W_z^T \tanh W_v V + (W_g h_T) 1^T \tag{3.37}$$

$$\alpha_t = softmax(z_t) \tag{3.38}$$

3.5.2 Objective Function

As similar to conventional image caption tasks, Adam optimizer reduces the cross-entropy loss, but the cross-entropy loss is defined differently from the previous methods.

$$\begin{aligned} L(\Theta) = -\sum_{t=1}^{T} log(p(w_t^*|\hat{j}, w_{1:t-1}^*) p(\hat{j}|w_{1:t-1}^*) 1_{w_t^*=w^{txt}} \\ + p(y_t^*, s_t^*|j_t, w_{1:t-1}^*)(\frac{1}{m}\sum_{i=1}^{m} p(j_t^i|w_{1:t-1}^*)) 1_{w_t^*=w^{vis}}) \end{aligned} \tag{3.39}$$

$1_{w_t^*}$ can be changed between text word and visual word. Therefore, during training,when the model deals with text word, $1_{w_t^*}$ is equal to w^{txt} and $log(p(w_t^*|\hat{j}, w_{1:t-1}^*)$ $p(\hat{j}|w_{1:t-1}^*)$ is enable. $p(y_t^*, s_t^*|j_t, w_{1:t-1}^*)$ and $(\frac{1}{m}\sum_{i=1}^{m} p(j_t^i|w_{1:t-1}^*))$ are enable, otherwise.

3.6 Unsupervised Image Caption

This chapter's previous methods were developed in the supervised learning aspect and reinforcement learning aspect using image-sentences pairs. In this section, one of the first unsupervised learning methods will be explained. Unsupervised Image Caption [12] model claims that creating these pairs causes too much work-load, and the description of the images might not have enough details for all images. Therefore, [12] suggests unsupervised methods without using any image-sentence pairs. However, since the model still needs to recognize the objects on the image, a visual concept detector is trained based on OpenImages Dataset. Finally, to create a relationship between image and sentence, image features and word embedding are mapped

to a common latent space. Thus, these pre-trained models and latent space become a bridge between image features and words.

3.6.1 Algorithm

The Unsupervised method [12] also utilizes CNN as encoder and RNN as decoder just as a typical image caption task. The model has two main components, which are generator and discriminator. These components are parts of the generative adversarial network. Firstly, the generator generates a caption for a given image. The architecture of the generator is almost the same as the decoder of [38]. The output of CNN's last layer is used as the initial value of the input of the first LSTM, which is defined as the generator. Then, LSTM starts to produce words. As for the discriminator, it evaluates the generated sentence. In this process, the second LSTM calculates a probability that the current sentence occurs. In brief, while the discriminator tries to understand that a given caption is from a text corpus or generator, the generator tries to trick the discriminator that the generated caption is from a text corpus. Therefore, the generator's reward is calculated as $r_t^{adv} = log(q_t)$. The generator learns how to create sentences with correct grammar for a given image feature using the adversarial network's components. The discriminator's loss function is computed as follows;

$$L_{adv} = -[\frac{1}{l}\sum \log(\bar{q}_t) + \frac{1}{T}\sum_{t=1}^{T} \log\left(1 - q_t\right)] \tag{3.40}$$

Even though none of the captions in the training dataset is used, objects on the image should be defined as words in the sentence. For this reason, a visual concept detector is utilized. (w_{cn}, κ_n) , which are concepts and probability of the concept, are calculated for every image. Besides, when the model produces a concept word ($w_t = w_{cn}$), it receives a reward due to the probability (κ_n) of the corresponding concept word.

$$r_t^c = \sum_{n=1}^{N_c} I(w_t = w_{cn}) * \kappa_n \tag{3.41}$$

Even though the model can build a link between image and sentence using concept reward, to add more semantic relationships, the model also maps both image feature and text feature to a latent space and tries to reconstruct both these features after

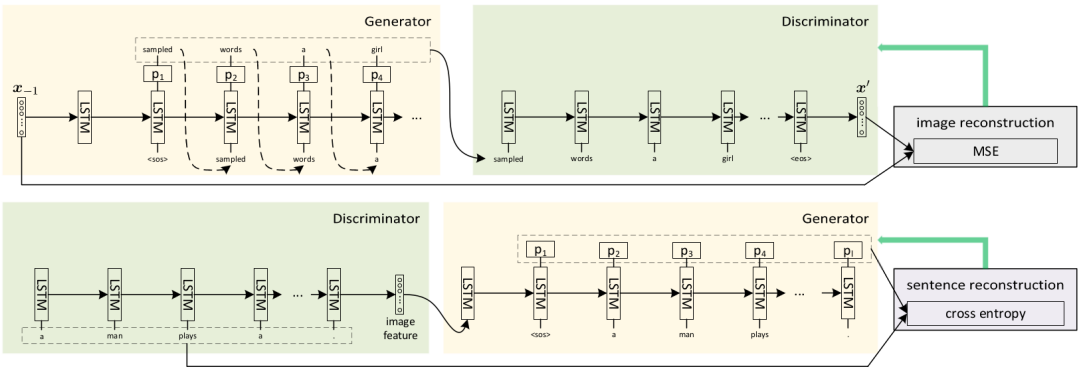

Figure 3.7: Image and Text Features Reconstruction from Unsupervised Image caption [12]

mapping them to the latent space. To check how reconstruction is successful, Norm2 is used. Below, $x^{'}$ represents generated image features.

$$L_{im} = ||v_{-1} - v^{'}||_2^2 \tag{3.42}$$

$-L_{im}$ is used as an image reconstruction reward r_t^{im} to update the generator's weights. In the same way, after reconstructing text features, the cross-entropy loss function is used to calculate reconstruction loss as follows;

$$L_{sen} = -\sum_{t=1}^{l} \log(p(w_t = \hat{w}_t | \hat{w}_t, ..., \hat{w}_{t-1})) \tag{3.43}$$

A policy gradient is utilized to compute the gradient for the reward function. Backpropagation, on the other hand, is only used for caption reconstruction. The total gradient of the generator is calculated as follows;

$$\begin{aligned} \nabla_\theta L(\theta) = -E[\sum_{t=1}^{n}(\sum_{w=t}^{n} \gamma^w (r_w^{adv} + \lambda_c r_w^c) \\ + \lambda_{im} r_w^{im} - b_t) \nabla_\theta log(w_t^T p_t)] + \lambda_{sen} \nabla_\theta L_{sen}(\theta) \end{aligned} \tag{3.44}$$

Where γ is decay factor, λ_c,λ_{im}, λ_{sen} are the weights of the terms. In addition, discriminator's loss function is computed as follows;

$$L_D = L_{adv} + \lambda_{im} L_{im} \tag{3.45}$$

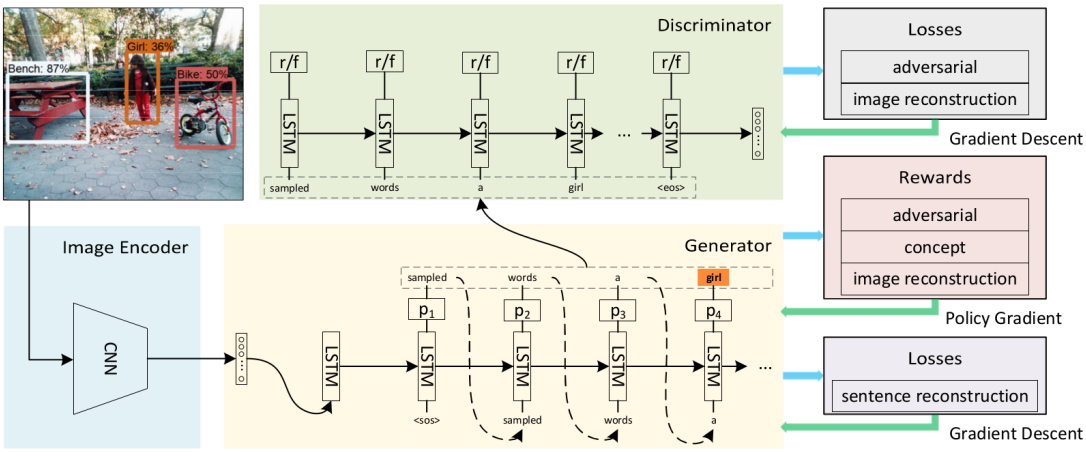

Figure 3.8: Unsupervised Image Caption Architecture from Unsupervised Image caption [12]

3.6.2 Initialization

Since training unsupervised methods need a lot of time, and the loss might be tremendous at the start point, some initialization methods are applied to the model to reduce this time and the loss. Therefore, derived captions for images are created to train the model in a supervised learning manner. Firstly, concepts on the image are detected. Then, two LSTMs are used to transform concepts into a sentence. In this LSTM, while the first LSTM tries to detect concepts in the sentence from the text corpus and transforms them into a latent space, the second LSTM receives this vector and use it to reconstruct original sentences. Then, derived captions are created using both the concept-to-sentence model and visual concept detector. Lastly, using these captions, images are trained as a supervised method.

3.6.3 Text-corpus

Text-corpus is prepared using Shutterstock. It is a website that includes sentences with details that describe the images. The names of the objects in MsCOCO are searched on the website to increase the similarity with the training data. While doing so, sentences with less than eight words are removed from text-corpus. In the end, 2.322.628 sentences are gathered.

3.7 Meshed Memory Transformer for Image Caption

Although there are some experiments on the Sequential Convolution Network, almost the whole algorithm utilizes LSTM (See Appendix A.3) to decode the image's abstract representation from latent space in Image Caption tasks since LSTM generally provides promising results for sequential processing. To connect words in a sentence, LSTM holds previous information in a memory cell. Thus, this previous information passes recursively to the next state. Moreover, it tries to learn what to remember or forget from previous information during training using forget gate. However, some experiments [30], [9] show us that LSTM cannot manage to create long dependency for long sentences using this memory cell mechanism. Therefore, rather than creating a recurrent network, the transformer utilizes a self-attention mechanism to link relationships between words. As for Meshed Memory Transformer for Image Caption [7], it is one of the first models that uses only a transformer to generate words and obtain state-of-the-art performance. Besides, global memory is added into the transformer as a learnable parameter to extract abstract information of the data set. Thus, the relationship between objects becomes related to more than one picture. In this section, the detail of the Meshed Memory Transformer will be examined.

In this model [7], the features of object regions on the image are extracted using Bottom-Up image features from Bottom-up and Top-Down attention [3] (See in section 3.4). Then, these are presented to the encoder of the transformer. The model [7] tries to understand the relationship between regions. It also decodes what is encoded in the encoder of the transformer. However, instead of only using the last layer, the decoder and encoder are connected at every layer. Thus, the decoder predicts which regions it should focus on.

In addition, similar to Self-critical Sequence Training for Image Captioning [35], Meshed-Memory transformer for Image Caption [7] also utilizes reinforcement learning after training with supervised learning. The CIDEr metric is, again, used as a reward function. Therefore, Meshed-Memory transformer for Image Caption [7] also uses REINFORCE with a baseline formula 3.26. However, rather than using Test module's output such as in Self-critical Sequence Training for Image Captioning [35], the model takes N number of reward results in Train Module and it uses average of

reward results as a baseline in REINFORCE with a baseline formula 3.26.

3.7.1 Algorithm

3.7.1.1 Encoder

The relationship between image features and words are extracted using self-attention mechanisms. This attention mechanism contains three values known as Query, Key, and Value. The intuition comes from propositional retrieving information tasks. Query vector is the whole abstract information. It is our abstraction of the whole target regions. Key vector is what we are looking for. Thus, $Q.K^T$ represents the similarity between Query and Key value. This value is scaled by the square root of the dimension of Key vector. Moreover, it is normalized by the soft-max function. Value is region information that we will use. The Value concept is very similar to Key. They might be equal in some cases.

$$Attention(Q, K, U) = softmax(\frac{QK^T}{\sqrt{D_k}})U \tag{3.46}$$

Overall, Weighted Average Value information is extracted. Q, K, and U values are calculated using different fully connected layers. D_k is the dimension size of K vector. Since all regions have a connection with each other, this event is called self-attention. The aim is to obtain the relationship between regions in the input.

Besides, which region should be paid propositional attention is computed at each encoder layer but to create a more complex relationship such as defining classroom by considering tables, blackboard, desk, e.t.c., the model tries to use prior knowledge by creating a memory concept. For this reason, additional empty blocks M_k ,M_u are defined in K and U. M_k and M_u can be defined as other learnable parameters.

In contrast to the traditional transform method, Attention, K, and U parameters are calculated as follows;

$$\begin{aligned} M_{mem}(X) &= Attention(W_q X, K, U) \\ K &= [W_k X, M_k] \\ V &= [W_u X, M_u] \end{aligned} \tag{3.47}$$

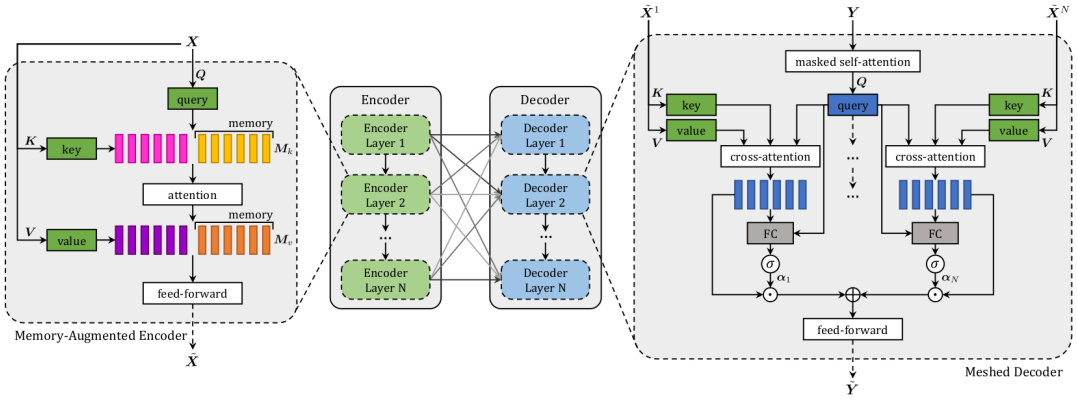

Figure 3.9: Meshed Memory Transformer Architecture from Meshed Memory Transformer for Image caption [7]

The output of the Attention mechanism is applied to the feed-forward network. W_q, W_k and W_u are weights matrices, X_i is one of the regions on the image, and σ is the ReLU activation function.

$$F(X)_i = W_u\sigma(W_uX_i + b_x) + b_u \tag{3.48}$$

All in all, a transformer encoder layer contains the following steps;

$$Z = AddNorm(M_{mem}(X)) \tag{3.49}$$

$$\hat{X} = AddNorm(F(Z)) \tag{3.50}$$

3.7.1.2 Decoder

Instead of utilizing just one layer representation, the decoder combines all encoder layers' attention using a fully-connected layer between encoder and decoder. A cross-attention mechanism is used to compute the attention weights. As for this time, query value is computed using a word embedding vector Y while Key and Value vectors are extracted using encoding regions $\hat{X}$. In brief, which layer of the encoding region is selected is based on the similarity between W_qY and $W_k\hat{X}$. Using this similarity, the weighted Value is determined.

$$C(\hat{X}^i, Y) = Attention(W_qY, W_k\hat{X}^i, W_v\hat{X}^i) \tag{3.51}$$

The weighted cross attention of decoder is calculated as follows;

$$\alpha_i = \sigma(W_i[Y, C(\hat{X}^i, Y)] + b_i) \tag{3.52}$$

$$M_{mesh}(\hat{X}, Y) = \sum_{i=1}^{N} \alpha_i . C(\hat{X}^i, Y) \tag{3.53}$$

In contrast to region features, words are examined sequentially. This means that $word_{t+1}$ cannot be used while generating $word_t$. So, the masked self-attention mechanism is applied, and the current word is used as the next input of the decoder.

$$Z = AddNorm(M_{mesh}(\hat{X}), AddNorm(S_{mask}(Y))) \tag{3.54}$$

$$\hat{Y} = AddNorm(F(Z)) \tag{3.55}$$

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, datasets ,which are utilized to train the image caption models, will be explained. Before interpreting any model, the metrics are introduced. Then, details of implementation are presented. Then, results of the metrics and possible drawbacks of the models will be examined.

4.1 Data set

In this section, two datasets, which are used for the above methods, will be explained. These two datasets are divided into three categories: train data, validation data, and test data. All categories have specific image ids to compare the algorithms fairly in the literature. Therefore, the images in the training, validation, and testing dataset are the same for all methods. This division is called Karpathy split in the literature. Moreover, training data is used to find proper values for a model's weights to accomplish the task. During training, to decide what weights are optimal, validation data is used. Since none of the images in the validation data are used to train the model, it can give an idea about the performance of the models. The weights which give the lowest validation loss or the highest score from a metric is selected as the best model. Cross-validation is not used in any model. After the training step is finished, test data is utilized to evaluate the model's best weights. Moreover, it gives the final results for the performance of a model. Therefore, it should only be used after training a model. Usage of test and validation data are very similar. However, test data cannot be used in the training step.

4.1.1 Microsoft COCO

The images in the MsCoco dataset [1] [5] are originally collected by Microsoft for image classification, semantic segmentation and object localization problem. The images are selected based on an image with an everyday scene that contains objects with contextual relationships. The data set has then become suitable for image caption tasks by adding five sentences for each image. These sentences must be associated with current events on the image. Subjective words, such as nicknames, were not used. The sentences must have only important details about the images. Consequently, the largest dataset for the whole scene image caption task was created. It contains 128k images and 80 different object categories.

Besides, specific images are selected from the dataset, and they are divided into three categories such as train (118k), validation (5k), and test(5k) to compare the model correctly.

4.1.2 Flickr30K

In contrast to the MsCOCO dataset, Flickr30k [2] [33] is designed especially for image caption tasks. Therefore, it has some additional benefits. Co-reference chains of words exist in the dataset, which means that there are similar concepts in the sentences. For instance, in 'A man with gauges and glasses is wearing a Blitz hat' and 'A man with pierced ears is wearing glasses and an orange hat.', 'gauges' and 'pierced' can be related to similar concepts. However, the captions of MsCOCO datasets are created independently. Besides, an object region on the image and corresponding region's noun in the sentences are linked to each other. This connection might help in the development of some new techniques. Also, the average number of words (12) in captions of Flickr30k is much more than the average number of words in captions of MsCOCO (10). So, it contains more details about the images. Similar to MsCOCO, Flickr30k also has everyday scenes. Finally, the images that will be used during the whole process are collected from the Flickr website. They are divided into categories based on Karpathy's split, which means 29k images for training, 1k images for vali-

[1] https://cocodataset.org
[2] http://bryanplummer.com/Flickr30kEntities/

dation, and 1k images for testing.

Most papers utilize MsCOCO because of its large size. However, in this thesis, the Flickr30k dataset is also used for training and testing models to exploit some advantages of its dataset and also to avoid the wrong interpretation of the models.

4.2 Evaluation Metrics

Metrics are utilized to evaluate generated captions. Since even for people describing the contents of any images changes from person to person, each metric focuses on different objectives. On the other hand, almost all metrics take advantage of similar concepts, such as precision and recall. While precision measures the model based on a model's result, recall measures the model based on the dataset's labels. True positive, false negative, and false positive should be known to understand these concepts more deeply. True positive is the positive values for both model and dataset's label. However, if the model predicts a positive label as negative, a false negative occurs. In contrast, if the model predicts a negative label as positive, a false positive occurs.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$
$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.1}$$

In the image caption task, Precision and Recall are refer to following formulas:

$$Precision = \frac{number\ of\ common\ words\ between\ candidate\ and\ reference\ sentence}{number\ of\ words\ in\ candidate}$$
$$Recall = \frac{number\ of\ common\ words\ between\ candidate\ and\ reference\ sentence}{number\ of\ words\ in\ reference} \tag{4.2}$$

On the other hand, the calculation of precision and recall might be changed based on metrics. The codes of the metrics except for BERT can be reached on the GitHub

link[3]. The code [4] is used to evaluate captions using BERT.

4.2.1 BLEU

The algorithm for BLEU score [31] is very similar to the calculation of precision. However, there is one noticeable difference. In precision, the words of a candidate sentence are counted in all reference sentences. On the other hand, in the BLEU score, the maximum number of words in any reference is taken into account. For instance, let's assume that a candidate sentence is 'the the the the the the the' and a reference sentence is 'there is a cat on the mat.' The simple precision is equal to the number of candidate words occurring in the reference sentence/Total number of candidate words. So, it is equal to 7/7 since all 7 'the' words are in the reference sentence. But the BLEU 1-gram score is 1/7 since 'the' occurs only one time in the reference sentence. Also, in literature, n-grams is selected from 1 to 4.

$$BLEU_{n-gram} = \frac{\sum_{n-gram\ \epsilon\ hyp} count_clip(n-gram)}{\sum_{n-gram\ \epsilon\ hyp} count(n-gram)} \tag{4.3}$$

However, there are some shortcomings in the score function. First of all, the BLEU score does not focus on the grammar of the sentence. BLEU 4-gram might reduce these drawbacks, but it is not entirely avoided. Secondly, it only takes exact words into account. For example, 'boy' and 'man' can be evaluated as different words regardless of their similar meaning.

4.2.2 ROUGE

In similar to the BLEU score, ROUGE [23] also examines the exact words of two given sentences. Moreover, while $Recall$ measures how many words there are originally in reference sentences, $Precision$ measures how many words are necessary. Formula 4.2 are utilizes to calculate these concepts.

The ROUGE score is calculated using F-measure to capture the relationship between

[3] https://github.com/ruotianluo/coco-caption/tree/ea20010419a955fed9882f9dcc53f2dc1ac65092
[4] https://github.com/Tiiiger/bert$_score$

$Precision$ and $Recall$. In the formula 4.4, ι is taken as 1.

$$F_{score} = \frac{(1 + \iota^2)\, Recall\ Precision}{Recall\ +\ \iota^2\ Precision} \tag{4.4}$$

There are different kinds of ROUGE scores. However, image caption tasks usually utilize ROUGE-L. In this score, the longest common subsequence is examined between two sequences. One of the advantages of ROUGE-L is that n-gram is chosen automatically during the calculation since it always chooses the longest common subsequences. But, the score counts only one subsequence. Therefore, if there are two equal-sized common subsequences, the score is calculated using only one of them. Thus, even though the candidate sentence is very similar to the reference sentence, the final score might be low. Additionally, similar to the BLEU score, the meaning of the words is also not taken into account.

4.2.3 METEOR

METEOR [1] score not only focuses on exact words like previous algorithms but also examines synonym words and the similarity between singular and plural words. Therefore, three different modules are used to capture the analogy between words. To reduce the difference between singular and plural words, all words are converted to their root version using a Snowball Stemmer (Porter 2001). Also, a synonym of a word is detected using Word Net Dataset. The number of common words is then used to compute precision and recall like the formula 4.2. The score of the best result is chosen as the METEOR score between a caption and reference sentences.

However, the METEOR score only examines single word matches. To consider the order of matches, a penalty concept is defined. Words in the sentences are divided into subsequences which are called chunks in the literature. The chunks should have the largest size for penalty calculation. F_{score} is calculated differently by taking ι as

three.

$$F_{score} = \frac{10 * Precision * Recall}{Recall + 9 * Precision}$$
$$Penalty = 0.5 * (\frac{numberofchunks}{numberofcommonunigramswords})^3 \tag{4.5}$$

Moreover, the final METEOR score is calculated as follows;

$$Score = F_{score} * (1 - Penalty) \tag{4.6}$$

4.2.4 SPICE

All metrics above are initially used for machine translation tasks. On the other hand, SPICE [2] is derived for image caption tasks. It aims to capture the semantic relationships of words. The computations are very similar to the METEOR score. Besides, synonym words and the relationship between singular and plural words are also taken into account. However, SPICE deals with word graphs. To do that, as a first step, linguistic dependencies are extracted between words in both candidate and reference sentences using Standford's pre-trained model in Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: ACL. (, 2003).

$$G(\mu) =< O(\mu), E(\mu), B(\mu) > \tag{4.7}$$

In the formula, $O(\mu)$ represents the name of the object, $B(\mu)$ represents attributes sets and, $E(\mu)$ represents the relationship between objects. For instance, for a sentence "A young girl standing on top of a tennis court." , $O(\mu)$ is (girl) and (court), $B(\mu)$ is (girl,young), (girl, standing) and (court, tennis), and $E(\mu)$ is (girl, on-top-of, court). In addition, $G(\mu)$ becomes (girl), (court), (girl, young), (girl, standing)(court, tennis), (girl, on-top-of, court).

$$Pr(\mu, \zeta) = \frac{|G(\mu)| \cap |G(\zeta)|}{|G(\mu)|}$$
$$Re(\mu, \zeta) = \frac{|G(\mu)| \cap |G(\zeta)|}{|G(\zeta)|} \tag{4.8}$$

After extracting tuples, precision and recall are computed like in METEOR score. Moreover, the final F_{score} is calculated by taking ι as 1.

$$SPICE(\mu, \zeta) = F_{score} = \frac{2 * Pr(\mu, \zeta) * Re(\mu, \zeta)}{Pr(\mu, \zeta) + Re(\mu, \zeta)} \tag{4.9}$$

4.2.5 CIDEr

Similar to METEOR and SPICE, before examining the words, the words are converted to their root version, such as converting stays and staying to stay. Then, CIDEr [37] calculates Tf-IDF values. While Tf gives credits to words based on how popular they are, Idf reduces the weights of common words since they convey less information. The $\Upsilon_k(\zeta_{ij})$ and $\Upsilon_l(\zeta_{ij})$ functions show the common number of n-gram which occurs in reference sentence.χ is the number of reference sentences. $|I|$ is number of word in the dictionary. μ_i and ζ_i candidate and reference sentences. Sigma is chosen as 6 in [37]. Overall Tf-IDF calculation for a given reference and candidate sentences is as follows;

$$\Psi_k(\zeta_{ij}) = \frac{\Upsilon_k(\zeta_{ij})}{\sum_{w_l \, \epsilon \, \Omega} \Upsilon_l(\zeta_{ij})} log(\frac{|I|}{\sum_{I_p \, \epsilon \, I} min(1, \sum_q \Upsilon_k(\zeta_{ij}))}) \tag{4.10}$$

CIDEr score can also be used for different n-grams selection. $Cider_n$ is computed as follows;

$$CIDEr_n(\mu_i, \zeta_i) = \frac{1}{\chi} \sum_j \frac{\Psi^n(\mu_i).\Psi^n(\zeta_{ij})}{||\Psi^n(\mu_i)||||\Psi^n(\zeta_{ij})||} Penalty \tag{4.11}$$

$$Penalty = e^{\frac{-(lenght\ of\ \mu_i - lenght\ of\ \zeta_{ij})^2}{2\ sigma}} \tag{4.12}$$

The overall score is computed like below;

$$CIDEr(\mu_i, \zeta_i) = \sum_{n=1}^{N} \frac{1}{N} CIDEr_n(\mu_i, \zeta_i) \tag{4.13}$$

4.2.6 Word's Moving Distance

The algorithm for WMD [21] initially compares two different documents. The intuition is to measure similar content between two sentences with different words. For

this reason, a pre-trained Word2Vec model is utilized. For each word, the word with minimum distance is found in another sentence. The negative logarithm of a cumulative Euclidean distance is defined as Word's moving distance score.

4.2.7 BERT

Like WMD, BERT score [10] also utilizes embedding vectors to capture similarities between sentences. However, this time, the words are mapped using BERT's pre-trained model. Then, Recall and Precision are calculated. Moreover, cosine similarities are computed rather than vector distances.

$$\begin{aligned} P_{Bert} &= \frac{1}{|w_\mu|} \sum_{w_{\mu i} \, \epsilon \, w_\mu} \max_{w_{\zeta j} \, \epsilon \, w_\zeta} w_{\mu i}^T w_{\zeta j} \\ R_{Bert} &= \frac{1}{|w_\zeta|} \sum_{w_{\zeta j} \, \epsilon \, w_\zeta} \max_{w_{\mu i} \, \epsilon \, w_\mu} w_{\mu i}^T w_{\zeta j} \end{aligned} \tag{4.14}$$

$\bar{X}$ and X are word embedding versions of both caption and reference sentence. Since word embedding vectors are normalized and $||X_i||$ and $||\bar{X}_j||$ are equal to 1, they are used not in the cosine similarity formula. Besides, to scale the score between 1 and 0, a re-scaling formula is used as follows;

$$\bar{R}_{base} = \frac{R_{Bert} - base}{1 - base} \tag{4.15}$$

In addition, $F_{measure}$ is also calculated as above formula by taking ι to 1 in equation 4.4.

4.3 Implementation Details

To make sure that the paper results are correct, the models are run once more on the computer. Ubuntu 16.04 and 18.04 operating systems are utilized. Nvidia GeForce GTX 1050 Ti and 1080 Ti graphic cards are used while both training and testing the models. Besides, most of the models are only trained in MsCOCO dataset because of its large size. However, since the dataset is initially created for object detection

and segmentation tasks and the drawbacks of the dataset might occur and cause criticizing wrong models, the flickr30k dataset is also used to compare the models to be more accurate. Words which appears at least five times in the sentences are used. In addition to that, the Beam Search algorithm is used for more proper captions. Beam Search is the algorithm that can find the N highest condition probabilities for words while generating a sentence. In the end, the algorithm presents N best captions.

4.3.1 Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

The algorithm (See section 3.2) was designed by Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. The code was written in Lua scripting language with Torch library. It is available in Github repository [5]. The captions of the model for both datasets are produced using pre-trained weights.

4.3.2 Self-critical Sequence Training for Image Captioning

Unfortunately, the original code of the algorithm (See section 3.3) is not presented by the author. But the code was written in Python scripting language with Pytorch framework by Ruotian Luo with the author's leading help. The code is available in Github repository [6]. Even though pre-trained weights are available for MsCOCO in the repository, the model was trained for both MsCOCO and Flickr30k dataset using Nvidia GeForce GTX 1050 Ti to make sure that implementation is correct.

4.3.3 Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

Pre-trained bottom-up image features for MsCOCO are used. They were extracted using Faster R-CNN object detection model that was trained on Visual Genome dataset. Similar to Self-critical, the PyTorch implementation of the paper used for Bottom-up and Top-Down attention model (See section 3.4) was not written by the author.

[5] https://github.com/jiasenlu/AdaptiveAttention
[6] https://github.com/ruotianluo/self-critical.pytorch

The code is available in the Github repository [7]. Also, differences from the original paper were indicated, such as using the ReLU activation function instead of Tanh. Therefore, the additional arrangements were removed. To train for Flickr30k dataset, the bottom-up features are obtained from "Stacked Cross Attention for Image-Text Matching" Github repository [8] in npy format. The model was trained and tested for both datasets using Nvidia GeForce GTX 1080 Ti.

4.3.4 Neural Baby Talk

The code [9] of the algorithm (See section 3.5) was presented with the original paper. It was written in Python scripting language with Pytorch framework. Besides, the pre-trained weights were utilized for datasets to obtain exact results on the paper. The code is available in the Github repository.

4.3.5 Unsupervised Image Captioning

The author presented the original code of the model (See section 3.6). Similar to the previous models, it was written in Python scripting language with the Tensorflow framework. Pretrained model is used to reproduce the same results on the paper. The code is available in Github repository [10].

4.3.6 Meshed-Memory Transformer for Image Captioning

The original code of the algorithm (See section 3.7) was used to obtain the results on the paper. The code[11] was written in Python 3.6 scripting language with Pytorch framework. The same pre-trained bottom-up features with Top-down attention are utilized. The code is available in the Github repository. For training and testing the Flickr30r dataset, the code was modified. It was trained using Nvidia GeForce GTX 1080 Ti.

[7] https://github.com/poojahira/image-captioning-bottom-up-top-down
[8] https://github.com/kuanghuei/SCAN
[9] https://github.com/jiasenlu/NeuralBabyTalk
[10] https://github.com/fengyang0317/unsupervised$_{c}aptioning$
[11] https://github.com/aimagelab/meshed-memory-transformer

4.4 Test Results

In this section, the results from code runs and results taken directly from the references will be examined for both datasets. The algorithms are explained in Chapter 4. To compare captions, metrics explained in Chapter 4.2 are utilized. The models are trained with datasets explained in Chapter 4.1. Unfortunately, the unsupervised image caption model (See methods 3.6) could not be trained for the Flickr30k dataset (See Chapter 4.1.2) because it needs a lot of training time and GPU memory. It is also the least successful method based on all metrics. Also, unsupervised methods usually need lots of samples. Therefore, the number of images in the Flickr30k dataset might not be enough for unsupervised image caption. Therefore, captions from the unsupervised image caption for the Flickr30k dataset are not examined. For the methods, the results can be found in the tables below.

Table 4.1: RESULTS BY VARIOUS EVALUATION METRICS FROM 6 MODELS WHEN TESTED ON MSCOCO DATASET WITH KARPATHY'S SPLIT. MODELS ARE IMPLEMENTED AS EXPLAINED IN SECTION 4.3

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	WMD	BERT
Self-Critical	74.9	57.87	43.9	31.63	25.3	54.2	105.1	18.8	21.7	61.59
Neural Baby	75.5	59.1	45.2	34.4	26.6	55.4	106.0	19.8	23.1	63.12
Meshed Memory	80.8	65.33	50.93	39.09	29.18	58.63	131.2	22.6	27.1	62.55
Meshed Memory wo bad ending	80.5	65.6	51.6	39.9	28.18	58.9	128.9	22.8	27.0	65.58
Unsupervised	41.0	22.5	11.2	5.6	12.4	28.7	28.6	8.1	8.6	33.91
Top-Down Attention	75.6	59.5	45.8	35.2	27.1	55.9	111.2	20.2	24.6	62.86
Adaptive Attention	73.8	57.5	43.7	33.1	26.4	54.7	105.1	19.3	23.3	62.47

Table 4.2: RESULTS TAKEN DIRECTLY FROM THE REFERENCES OF 6 MODELS ON MSCOCO DATASET WITH KARPATHY'S SPLIT

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	METEOR	ROUGE-L	BERT	WMD	SPICE
Self-Critical	-	-	-	31.9	106.3	25.5	54.3	-	-	-
Neural Baby	75.5	-	-	34.7	107.2	27.1	-	-	-	20.1
Meshed Memory	80.8	-	-	39.1	131.2	29.2	58.6	-	-	22.6
Unsupervised	41.0	22.5	11.2	5.6	28.6	12.4	28.7	-	-	8.1
Top-Down Attention	77.2	-	-	36.2	113.5	27.0	56.4	-	-	20.3
Adaptive Attention	74.2	58.0	43.9	33.2	108.5	26.6	-	-	-	-

As you can see, the results from the codes in Table 4.1 and the papers in Table 4.4 are very close to each other. The difference occurs when the models are re-trained. Usually, the researchers do many experiments for their model, and they put the best results in their paper. Therefore, retraining the models might cause to have less successful results. Another reason is that Top-Down [3] (See Chapter 3.4) and Self-critical [35]

(See Chapter 3.3) were not implemented by their authors. Thus, there might be little changes in the code.

When BLEU scores (See Chapter 4.2.1) are examined in Table 4.1. Meshed-Memory model (See Chapter 3.7) has the highest BLEU score. In other words, the number of exact n-grams between reference sentences and generated sentences from the Meshed-Memory model is the greatest among the other models. Even though the results from Top-Down attention (See Chapter 3.4) and Neural Baby Talk (See Chapter 3.5) are similar, Top-Down attention seems a little bit more successful. They both utilize Faster-RCNN for object detection. However, using class labels might give less score to the Neural Baby model for comparing exact words. After that, the results from Self-critical learning (See Chapter 3.3) and Adaptive attention (See Chapter 3.2) are also very close. Self-critical learning is slightly better than Adaptive attention. This means an attention mechanism might not be enough, and reinforcement learning methods show us better results to produce exact n-grams. The last model is the unsupervised image caption. Since reference sentences are not used in Unsupervised Image Caption (See Chapter 3.6), it is no surprise that the exact common n-grams are less than other models. As mentioned before, BLEU score only focuses on exact n-grams. Other qualities, such as synonym words, are not taken into account. Thus, only using BLEU score is not enough to compare the models.

Similar to BLEU score, ROUGE-L (See Chapter 4.2.2) also examines the exact n-gram. However, instead of using a specific n-gram, it focuses on the Longest Common Subsequence. If the number of the common subsequence is bigger, ROUGE-L score becomes greater. Besides, it also computes an F score using precision and recall. Moreover, Meshed-Memory [7], again, has the greatest score. It means common exact n-grams are more than four. After that, Top-Down attention [3] is a little bit better than Neural Baby [26]. Attention LSTM might give better captions (See details in 4.4.3). In contrast to BLEU score, however, Adaptive attention [25] has greater score than self-critical [35]. This means in this harmonic weighted between precision and recall, Self-critical [35] is less successful but very close to Adaptive attention [25]. On the other hand, again, Unsupervised Image Caption [12] is the worst model based on ROUGE-L.

In contrast to previous metrics, METEOR (See Chapter 4.2.3) does not only focus on the exact word. It removes the differences between singular and plural words, and it also examines synonyms words. As you can see, in the table Meshed-Memory [7] has the highest score. Then, Top-down attention model [3] comes. The scores for Neural Baby [26] and Adaptive attention [25] are very close in contrast to the above scores. Interestingly, Neural Baby Talk [26] also examines synonyms and singular-plural words while producing a sentence. However, the METEOR score of Neural Baby Talk [26] is not significant. Therefore, the additional network for synonyms and singular-plural words in Neural Baby Talk [26] might not be very useful. Then, Self-critical learning [35] becomes the fifth successful model. Checking additional information might not affect performance of Unsupervised image caption model [12].

CIDEr score (See Chapter 4.2.5) examines the root of the words. Moreover, it calculates weights to decide whether a word is important using inverse document frequency. The less important words such as 'a', 'with' or 'the' obtain fewer weights since they do not hold significant meaning. These weights are computed for all words in the word dictionary using all reference sentences in the datasets. In this manner, the order of the performance of the model is similar to previous metrics. Meshed-Memory [7] has the greatest score, and the Unsupervised image caption model [12] has less successful captions. Top-down attention model [3] is very successful compared to Neural Baby Talk [26] according to CIDEr score. Thus, it might be said that words in Top-Down attention might be more informative than words in Neural Baby Talk [26]. Interestingly, Self-critical [35] and Adaptive Attention [25] have the same score for CIDEr.

Furthermore, SPICE (See Chapter 4.2.4) is the only model that gives the relationship between objects and attributes a credit. Therefore, SPICE might tell more information about similar contents between reference sentences and caption. The order does not change too much. Captions of Meshed-Memory [7] has a more common relationship with reference sentences. On the other hand, Adaptive Attention [25] is dramatically better than Self-critical [25] according to SPICE score.

WMD (See Chapter 4.2.6) is a simple score that tries to compare sentences based on word embedding distance. In word embedding space, every word has a correspond-

ing vector representation. According to WMD, the minimum distance between sentences give the maximum score. Thus, the order of performance, again, is the same. Meshed-Memory [7] is the model that has the lowest distance to reference sentences. Adaptive attention's [25] score is very close to Neural Baby [26]. On the other hand, it significantly exceeds Self-critical's [35] score. Unsupervised Image Caption [12] , on the other hand, has the greatest distance to reference sentences among others.

BERT (See Chapter 4.2.7) is the newest metric among others. It also utilizes word embedding vectors. One of the main differences to WMD, embedding vector is computed using a BERT Network. Also, F-score is calculated with precision and recall values. Surprisingly, according to BERT score, the Neural Baby model [26] is much better than the other. Meshed-memory [7] model is the third successful methods. The Adaptive attention [25], again, has higher score than Self-critical [35].

The details of Meshed memory with no bad endings is explained in Chapter 4.4.3.

Table 4.3: RESULTS BY VARIOUS EVALUATION METRICS FROM 6 MODELS WHEN TESTED ON FLICKR30K DATASET WITH KARPATHY'S SPLIT. MODELS ARE IMPLEMENTED AS EXPLAINED IN SECTION 4.3

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	WMD	BERT
Self-Critical	65.0	44.6	30.4	20.3	18.2	43.9	43.8	11.9	13.1	52.5
Neural Baby	69.0	51.0	37.3	27.0	21.6	48.2	56.8	15.6	16.3	51.0
Meshed Memory	72.7	55.8	41.6	30.8	21.9	49.9	64.9	16.2	18.1	58.6
Unsupervised	-	-	-	-	-	-	-	-	-	-
Top-Down Attention	69.5	52.0	38.3	28.4	21.2	48.3	57.7	15.8	17.0	55.6
Adaptive Attention	67.6	49.2	35.4	25.5	20.2	47.1	49.7	13.7	15.0	53.5

Table 4.4: RESULTS TAKEN DIRECTLY FROM THE REFERENCES OF 6 MODELS ON FLICKR30K DATASET WITH KARPATHY'S SPLIT

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	METEOR	ROUGE-L	BERT	WMD	SPICE
Self-Critical	-	-	-	-	-	-	-	-	-	-
Neural Baby	69.0	-	-	27.1	57.5	21.7	-	-	-	15.6
Meshed Memory	-	-	-	-	-	-	-	-	-	-
Unsupervised	-	-	-	-	-	-	-	-	-	-
Top-Down Attention	-	-	-	-	-	-	-	-	-	-
Adaptive Attention	67.7	49.4	35.4	25.5	53.1	20.4	-	-	-	-

In contrast to Table 4.2, table 4.4 does not have many scores. Most of the papers in the literature utilize only MsCoco dataset because of its large size. However, as mentioned in Chapter 4.1.2, the Flickr30k dataset was specially designed for image caption task. Besides, sentences in the dataset contain more words. This helps the model link more relationships between objects and words. Also, Neural Baby talk

[26], and Adaptive attention [25] are only two models that are trained with Flickr30k dataset in original the papers. Moreover, the order of the results ,when the code is implemented, is very close to the results provides in the papers.

All metrics above are, again, used to evaluate captions. The order of the successful models is very similar to the order obtained for MsCoco. However, at this time, the result of Adaptive attention model [25] is substantially greater than Self-critical model [35]. The reason might be Adaptive attention model might deal with long sentences better than Self-critical. Since Self-critical does not use spatial attention and LSTM might lose the image's information in the last step (See details in Chapter 4.4.3). According to BERT score, Meshed-memory [7] score is slightly greater than Neural baby model [26] for flickr30k dataset.

Table 4.5: NUMBER OF DIFFERENT CAPTION FROM THE 6 MODELS FOR BOTH DATASETS

	Self-critical	Adaptive	Unsupervised	Meshed Memory	Top-down	Neural Baby
MsCoco	2835	3300	3231	3918	2428	3431
Flickr30k	736	911	-	961	953	955

The number of different captions is shown in Table 4.5. This might not be the best way to measure the novelty of the captions. However, it might help to compare. As you can see, generated captions from Meshed-Memory model [7] are very different from each other for both datasets. On the other hand, the captions can be ended with bad endings such as 'with a','on a' (See Chapter 4.2). If bad endings are removed by hand, this number might be decreased. Neural Baby talk [26], on the other hand, produces non-identical captions than the other methods except Meshed-Memory [7] as it is claimed in the paper [26]. According to the table, Top-down attention [3] tends to generate the same caption for the MsCoco dataset. The reason might be Top-down attention [3] generally focuses on object features rather than the whole scene. Thus, two images with the same objects might cause an identical caption. Since the Unsupervised image caption model [12] does not utilize any reference sentences during creating captions, it cannot copy the reference sentences. As mentioned above, Self-critical [35] learning only uses the image feature initially. In the last step of Self-critical [35], the image's information might be lost. Therefore, looking at the image only once causes similar captions. On the other hand, since adaptive attention [25]

focuses on different parts of the image, the caption might be non-identical. On the other hand, the number of identical captions is very low in the Flickr30k dataset. The reason might be since reference sentences are longer in the Flickr30k dataset, the models might learn to generate longer captions. Increasing the number of words might decrease the possibility of generating the same caption. In addition to that, the number of test images in Flickr30k is also less than the number of test images in MsCoco. Therefore, test images in Flickr30k might be more dissimilar.

4.4.1 Comparison of Evaluation Metrics

Most of the evaluation metrics, which are used in the thesis, are well-known metrics. Moreover, they are utilized in many papers in the literature. However, they are not perfect. For instance, many metrics such as BLEU (See Chapter 4.2.1), ROUGE-L (See Chapter 4.2.2), and CIDEr (See Chapter 4.2.5) examine exact words or word groups. The number of words that BLEU score focuses on can be changed based on n-gram. Since it does not check any grammar rule, it tries to reduce this drawback by examining word groups. Moreover, the ROUGE-L score also looks into the longest word groups. Therefore, generally, the evaluation order of ROUGE-L and BLEU-4 scores are very similar. In addition to that, CIDEr score also uses tf-idf. Idf gives weights a word based on its importance. However, since the importance of a word is decided according to the number of usage of the words, it is possible that an unpopular and unimportant word might have a high weight. In addition, since they examine exact words, a sentence which have the same meaning as reference sentences might not obtain very good results from above metrics when it utilizes different words with reference sentences. Besides, METEOR (See Chapter 4.2.3) also focuses on exact words. However, it also examines synonym words and the root of the words, but this similarity is obtained based on predefined dictionaries such as Word Net Dataset. Therefore, the mapping of synonym words is also limited.

Similar to METEOR, SPICE (See Chapter 4.2.4) also looks into a synonym of words. In addition, it basically creates groups of words for a given sentence based on linguistic dependencies of words. However, wrong groups of the words are also possible. Furthermore, it might change the final score. And, metric might miss-evaluate cap-

tion.

As mentioned before, WMD (See Chapter 4.2.6) and BERT (See Chapter 4.2.7) deal with word embedding vectors instead of string words. Thus, during comparing sentences, the similarity between words does not depend on any predefined vocabulary. On the other hand, WMD extracts embedding using a Shallow network and its word embeddings are not contextual word embedding. It means that the same word which has two different meanings might have the same word embeddings. Therefore, it might mislead the evaluation. Furthermore, BERT has contextual word embeddings and it also utilizes tf-idf similar to CIDEr score.

In [18], all metrics, that are discussed except BERT, are examined and metrics are correlated with human judgment in Flickr8k and Composite datasets. As a conclusion of the study, WMD, METEOR, and SPICE have the best results based on correlation analysis. In addition, in [35], CIDEr is used as a reward function because it increases results of all metrics. The reason might be CIDEr score is qualifier to understand differences between bad captions and perfect caption than others such as shown in Table 3 in [18]. On the other hand, sentences with bad endings are not taken into account in [18]. Our results shows that metrics are not very sensitive to bad endings except for BERT and CIDEr metric. While bad endings increase CIDEr score's results, they decrease BERT score's results according to Table 4.1.

4.4.2 Captions from the models

Since evaluating captions is a significantly challenging task, to be more precise, some of the arbitrary examples are shown in this section.

Figure 4.1: 90476 image number from MsCOCO dataset

Reference sentences

A large number of boats that have been beached.

The ships are all docked on the beach by the water.

A group of boats on top of wet sand.

A group of boats in the sandy area of a beach.

Several beached boats on the sand with orange balls hanging over the sides.

Captions

A bunch of boats that are sitting in the water

(Adaptive)

A group of boating that are sitting in the sand

(Neural)

A group of boats are sitting in the water

(Self)

Fishing boat on the beach at sunset

(Unsupervised)

A group of boats sitting next to each other

(Top Down)

A group of boats are parked on the beach

(Meshed Memory)

Figure 4.2: 182245 image number from MsCOCO dataset

Reference sentences	Captions
Man sitting at a small table behind a pizza.	A person sitting at a table with a pizza (Adaptive)
A man sitting at a table with lots of food in front of him.	A man sitting at a table with a plate of pizza (Neural)
Friends about to enjoy fresh pizza and coffee at a restaurant.	A man sitting at a table with a pizza (Self)
The Asian man is deciding what to eat on the plate.	A cup of coffee on a wooden table (Unsupervised)
A man sits at the table with a large pizza on it.	A man sitting at a table with a pizza on it (Top Down)
	A man sitting at a table with a pizza on a (Meshed Memory)

Figure 4.3: 271240 image number from MsCOCO dataset

Reference sentences	Captions
A stop sign and street sign encased in snow and ice.	A stop sign in front of a snowy mountain (Adaptive)
A couple of street signs sticking out the side of a ski slope.	A stop sign covered in snow on a snowy surface (Neural)
A stop sign and street sign encased in a wall of snow.	A stop sign on the side of a snow (Self)
A stop sign is buried in a large pile of snow.	No smoking sign on a white background (Unsupervised)
A large pile of snow with a stop sign and a street sign poking out.	A stop sign covered in snow covered in snow (Top-Down)
	A stop sign covered in snow on a street (Meshed Memory)

Figure 4.4: 384981 image number from MsCOCO dataset

Reference sentences	Captions
A group of people gathered to celebrate a birthday.	A man sitting at a table with a birthday cake (Adaptive)
The cake is decorated for the family celebration.	A group of people sitting around a table with a cake (Neural)
People gather around and sit at a table around a birthday cake.	A group of people sitting at a table with a cake (Self)
A kid is sitting ion front of a cake.	A beautiful girl is sitting on a birthday cake with a cake (Unsupervised)
A woman sitting in front of a cake on a table.	A woman sitting at a table with a cake on it(Top-Down)
	A group of people sitting around a table with a cake (Meshed Memory)

Figure 4.5: 451872 image number from MsCOCO dataset

Reference sentences	Captions
Several elephants are in a habitat as heads are in the foreground.	A group of elephants standing next to each other (Adaptive)
A small gray elephant standing in an exhibit at a zoo.	A group of elephants standing around in a zoo (Neural)
People are watching four elephants in a zoo.	A group of elephants standing next to each other (Self)
Several elephants in zoo enclosure with onlookers watching.	Elephant bull in the kruger national park , south africa (Unsupervised)
An elephant in a zoo stands in front of the crowd.	A group of elephants standing next to each other(Top-Down)
	A group of elephants in a zoo (Meshed Memory)

Figure 4.6: 507065 image number from MsCOCO dataset

Reference sentences	Captions
A little boy sitting on a wooden bench eating half a sandwich.	A young boy is eating a hot dog (Adaptive)
A small boy with a green shirt is eating a sandwich.	A young boy eating a piece of cake (Neural)
A young boy sitting on a bench with a sandwich.	A young boy sitting at a table eating a cell phone (Self)
A little boy holding half a sandwich in each hand.	Food , people and people concept - happy man and woman (Unsupervised)
A boy is holding a sandwich up to his face.	A young boy eating a piece of cake (Top-Down)
	A young boy eating a piece of cake with a (Meshed Memory)

Figure 4.7: 175556963 image number from Flickr30k dataset

Reference sentences	Captions
A group of spectators sitting in lawn chairs in the street facing a building and drinking beer.	A crowd of people are gathered around a crowd of people (Adaptive)
People sitting around in a semi circle all looking in the same direction.	A group of people sitting on a sidewalk (Neural)
A large group of people of various ages and genders sit outside together.	A group of people are sitting on a street (Self)
People sit in green chairs at tables in the street.	A group of people are sitting on a sidewalk (Top-Down)
Some of the people in the crowd are having a drink.	A large group of people are sitting in the street(Meshed-Memory)

Figure 4.8: 1798209205 image number from Flickr30k dataset

Reference sentences	Captions
A blond woman pours a glass of wine in a dim restaurant.	A woman in a white shirt is sitting at a table in a restaurant (Adaptive)
A woman in a restaurant pouring wine into a clear glass.	A woman in a white shirt is pouring a drink (Neural)
A blond-haired woman is pouring drinks at a bar.	A woman in a white shirt is sitting at a table (Self)
A woman with a tag is pouring red wine in a glass	A woman in a white shirt is sitting in a restaurant (Top-Down)
A lady in the kitchen is pouring wine.	A woman is pouring a drink into a glass (Meshed-Memory)

Figure 4.9: 2872099696 image number from Flickr30k dataset

Reference sentences	Captions
White man with a green shirt and denim pants with a hammer and chisel, creating some sort of artwork.	A man in a gray shirt is sitting on a stone bench (Adaptive)
A man with chin-length light hair works on a sculpture.	A man in a green shirt is sitting on a cart (Neural)
A man creates a sculpture with a hammer and chisel.	A man in a blue shirt is working on a skateboard (Self)
A man pounds on an artwork perched on a cart.	A man in a gray shirt and jeans is working on a sculpture (Top-Down)
A man is swinging a tool at his sculpture.	A man working on a sculpture (Meshed-Memory)

Figure 4.10: 179828434 image number from Flickr30k dataset

Reference sentences	Captions
A little boy is wearing a blue and white shirt while holding a toy animal that is either a crocodile or an alligator.	A little girl in a white shirt is holding a yellow stuffed animal (Adaptive)
A small child wearing a blue and white t-shirt happily holding a yellow plastic alligator.	A little girl in a white shirt holding a yellow toy (Neural)
A little boy in a light blue shirt playing outside with a toy crocodile.	A little girl in a blue shirt is playing on a street (Self)
A young smiling child holds his toy alligator up to the camera.	A little girl in a blue shirt is sitting on the ground with a stuffed animal (Top-Down)
A little boy holding a baby reptile.	A young girl is holding a banana (Meshed-Memory)

4.4.3 Analyzing details of the models

Algorithms of the models are explained in Chapter 3 and the results of the models based on metrics are shown in Table 4.1 and Table 4.3. The image caption is far more difficult than object classification and machine translation tasks. First of all, it is very challenging to explain images since there is no strict way to give information about

images, even for a human. Thus, more than one sentence for each image is needed. In addition to that, the sentences must be clear,must have all important details, and must be in present tense. Moreover, when one compares the results between MsCoco dataset (See Chapter 4.1.1) and Flickr30k (See Chapter 4.1.2) dataset, it can be said that the number of samples clearly increase scores of the models as in shown Table 4.1 and Table 4.3.

However, creating reference sentences is a very challenging task because the way of describing images might be changed even for a human. Therefore, details in the sentences might change too. A model might not utilize an object's name in the caption if this name is not in the reference sentences. Thus, Unsupervised learning for the whole scene image caption task is needed to be improved. On the other hand, the Unsupervised image caption model [12] (See Chapter 3.6) does not show very promising results. The author in the paper [12] claims that the main reason is that training the model without reference sentences creates dissimilar sentences to reference sentences. Scores from metrics consequently measure less similarity. Thus, the author's claim might be right. However, when one examines the caption from [12], there are still unrelated sentences such as in Figure 4.6. There are many reasons for this inappropriate caption. First of all, [12] does not have any attention mechanism. Creating a relationship between objects and words might increase in generating proper words. Secondly, dealing with the longest sentences might be difficult for LSTM to manage because LSTM might lose the information of previous states at the very last step. Therefore, LSTM might not learn to reconstruct image features for long sentences. Also, a transformer network might be used instead of LSTM. According to [36], a transformer network is better to hold long dependency than LSTM. Thirdly, [12] needs too much GPU memory and too much time. This prevents doing more experiments for new algorithms.

As mentioned above both Self-critical [35] (See Chapter 3.3) and Meshed-Memory network [7] (See Chapter 3.7) are optimized using Reinforcement learning after training using supervised learning. They both use CIDEr metrics. However, according to [27], while increasing the length of a caption, CIDEr metric can cause bad endings at the end of the sentence such as 'with a' or 'on a' (Figure 4.2 and Figure 4.6). For example, when 'a young boy eating a piece of cake' is evaluated,CIDEr score is 0.6561.

On the other hand, the score of 'a young boy eating a piece of cake with a' is 0.7044. This might happen when a reward function is not correctly designed in reinforcement learning. It is called reward hacking in the literature. Therefore, it can be said that CIDEr score is not adequate to be a reward function for models. In addition to that, Table 4.5 shows that [35] cannot generate different sentences. The reason might be that image features are only used at the beginning of the LSTM steps.

According to BERT score, Meshed Memory model [7] is the third successful model for MsCoco dataset [5]. On the other hand, the model is the most successful method for Flickr30k dataset [33]. In addition, as mentioned above, using CIDEr score as a reward function can cause bad endings for Meshed memory model [7]. Therefore, when captions with bad endings are counted, the trained model for MsCoco dataset has 1742 captions with bad endings out of 5000 while the trained model for Flickr30k dataset has 0 out of 1000. When bad endings in the captions are removed for the model ,which is trained for MsCoco dataset, by hand, it is clear that CIDEr score dramatically falls when BERT score is significantly increased. This supports the idea that CIDEr score causes bad endings. Moreover, BERT score can understand better the difference between a caption with a bad ending and a caption with no bad ending.

Furthermore, Meshed-Memory model [7] (See Chapter 3.7) utilizes scale dot attention. Scale dot attention helps the model measure similarity between object feature and word. However, in the transformer's decoder, [7] always calculates the similarity score even though there is no relationship. This problem of transformer is also mentioned in [15]. Moreover, Meshed-Memory transformer [7] always looks at image features for every word. Thus, the model computes similarity score between image features of an object and non-visual words. Adaptive attention[25] (See Chapter 3.2) shows how focusing on previous words for non-visual word helps to improve the performance of its model.

Adaptive attention [25] is very close to traditional approach such as [38], [41]. Nevertheless, Top-down attention [3] (See Chapter 3.4) and Neural Baby [26] (See Chapter 3.5) show that attention LSTM and object features from Faster-RCNN increase the scores of the model.

Moreover, designs of LSTM and image feature extraction in Top-Down Attention [3]

and Neural Baby Talk model [26] are alike. The object features are extracted from Faster-RCNN. Nevertheless, while Top-down attention [3] examines global image features and object features from Faster-RCNN, Neural baby talk [26] also utilizes a feature map of the last layer of CNN. Therefore, [26] has more information about an image. This might be the reason that [3] has a lot of exact sentences in Table 4.5 for MsCoco dataset. The images with similar objects cause the same words in the captions. For the Flick30k dataset, on the other hand, the number of exact sentences is very low. The reason might be that reference sentences in Flickr30k have more nouns, and the test sample size is significantly less than the test samples in MsCoco. Thus, encountering similar images is less possible. According to Table 4.5, Neural Baby Talk [26] is second model which generate different sentences. This result matches with claims of the author of Neural Baby Talk [26]. Another differences between Top-down attention [3] and Neural Baby Talk [26] that in Top down attention [3], attention LSTM takes previous hidden state of Language LSTM as one of its input. This might help attention LSTM keep track of language LSTM.

In addition, Meshed-Memory [7] also utilizes Bottom-up image features from Top-down attention paper [3]. However, Table 4.5 shows that Meshed Memory transformer [7] can generate distinctive captions. The reason might be that Meshed Memory transformer [7] adds global memory in the transformer's encoder. Thus, the relationship between objects in the image might not restrict to only one sentence. Therefore, it can be said that using prior knowledge might help to produce novel sentences.

Despite the bad endings, Meshed-Memory Transformer for Image Caption [7] can be considered as the best model among others. As mentioned earlier, one of the reasons is that transformer is better to hold long dependency between words than LSTM thanks to the self-attention mechanism. On the other hand, it is not the only reason. Transformer architecture in [36] is published in 2017. However, [7] is one of the first models which only used transformer for text decoder and reached state-of-the-art performance. The obvious difference between the original transformer and Meshed-memory transformer is global memory that helps the network to understand the relationship between objects. Furthermore, Meshed-memory transformer utilizes 3 layers while the original transformer network has 6 layers. Reducing the number of layers might help the network to avoid overfitting. In addition to that, Meshed-

memory transformer connects the encoder and the decoder at every layer rather than just one layer. Therefore, one layer in decoder might utilize the outputs of all encoder layers when it needs.

4.4.4 Future Work

In this Chapter, possible additions to the models will be explained.

As explained in Chapter 4.4.3, using CIDEr metric as a reward function can cause reward hacking problem. In addition to that, BERT score gives a low score when it encounters a sentence with a bad ending. Therefore, these two different metrics might be used together as a reward function. In this way, BERT score might limit the lousy effect of the CIDEr score.

In Chapter 3.2, the benefit of the visual sentinel is explained. The same mechanism might be added to Meshed Memory Transformer for Image Caption model [7] since the model always calculates a similarity score between object features and words.

In Chapter 3.7, advantage of transformer is explained. Moreover, Table 4.1 and Table 4.3 supports that idea. Therefore, for Unsupervised Image Caption [12], transformer network might provide better solution than LSTM.

CHAPTER 5

CONCLUSION

5.1 Conclusion

The purpose of this thesis is to compare algorithms,which make outstanding contribution to the literature, in the whole scene image caption task. All six models make outstanding contributions to the literature. However, focusing on the details of the models and test results makes it easier to develop further improvements. Many surveys in the literature examine only the traditional approaches such as [38] and [41] rather than the methods with new perspectives. In this thesis, this analysis and comparison of various methods is completed.

In addition, training models with more than one data set can help to understand the behavior of the models better. In this thesis, image-sentence pairs from MsCoco [5] and Flickr30k [33] (See Chapter 4.1) are selected as data sets. MsCoco usually is used in the image caption task because of the large number of image-sentences samples. Since data sets are prepared by human, it might have some flaws, and it might fail algorithms. To avoid this problem, the models are also trained with the Flickr30k data set. Meshed-Memory [7], Bottom-up and Top-Down attention [3] and Self-critical [35] papers does not have metrics scores for Flickr30k Data sets. Therefore, one of the contributions of this thesis is to show metric results for Flickr30k Data sets.

In this thesis, Adaptive attention [25], Self-critical [35], Bottom-up and Top-down attention [3], Neural Baby Talk [26], Meshed Memory model [7] and Unsupervised Image caption [12] methods (See Chapter 3) are chosen to compare. As mentioned above, all these methods have their perspectives to solve the image caption problem.

First of all, [25] creates a visual sentinel concept that can be considered abstract information of previous words. Moreover, [25] utilizes it as one of the activation grid from the image feature map. Secondly, [35] , on the other hand, adds optimizing network's weights with a metric. Thirdly, [3] does not rely on activation grids of the last layer of CNN to utilize features of the objects. Faster-RCNN detects objects and extracts features of the objects. Fourthly, [26], on the other hand, tries to develop novel sentences using an additional object detector. Fifthly, [12] is one of the first unsupervised methods in image caption task. Even though it cannot reach the state of the art performance, it brings new aim to the literature. [7] is also one of the first methods which do not utilize LSTM as a decoder.

Understanding the performance of the image caption models is a very challenging task because there is no straightforward way to explain the content of the image. The details of the contents can be changed from person to person. In the literature, metrics (See Chapter 4.2) evaluates captions from the models. Also, since metrics for image caption tasks are not perfect, additional metrics such as WMD and BERT are utilized to evaluate captions for both data sets in this thesis.

Test score from metrics are shown in Chapter 4.4. The order of test scores of the code and the paper for MsCoco dataset [5] are very close. Since not all model used Flickr30k [33], the comparison between the code and the paper is not possible. Furthermore, When the score results are examined for both data sets, it is clear that Meshed Memory [7] outperforms the others. On the other hand, Unsupervised image caption task [12] is the least successful model among others. Moreover, although metric's results of Neural Baby Talk [26] is very close to Top-down attention [3], [3] is generally second successful model according to metrics. Then, Adaptive attention [25] comes after Neural Baby Talk [26]. Besides, self-critical [35] is the fifth successful model based on the metrics.

In spite of the great contribution to the literature, these models also bring some disadvantages. First of all, unsupervised image caption [12] cannot utilize any attention mechanism. Therefore, it might not link strong relationships between objects and words. In addition, needing too much time and GPU memory for training prevent doing different experiments on the model [12]. Secondly, Meshed memory [7] is the

best model according to metrics. However, it calculates similarity scores for objects and non-visual words such as 'the', 'of' etc. Thirdly, Bottom-up and Top-down attention [3] generates exact sentences according to Table 4.5. Thus, [3] might not produce different sentences when it encounters images with similar objects. Fourthly, Neural Baby Talk [26] focus on creating more novel sentences. On the other hand, using the object labels of the object detector might decrease the metrics score if this label does not in any reference sentences. Fifthly, Adaptive attention [25] utilizes one LSTM to calculate attention and words. According to test results of [26] and [3] in Table 4.1 and Table 4.3, distinguishing LSTM for attention and language might increase the metrics results of the models. Lastly, Self-critical learning [35] utilizes a reward function to train a model using reinforcement learning. However, the Cider metric is not adequate to evaluate models. Therefore, a reward hacking problem might occur. Details of the analysis are explained in Chapter 4.4.3.

REFERENCES

[1] A. Agarwal and A. Lavie. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Third Workshop on Statistical Machine Translation, pages 115-118,Columbus, Ohio, USA. Association for Computational Linguistics.*, 2008.

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision and Pattern Recognition*, 2016.

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[4] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, and T. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[5] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. In *Computer Vision and Pattern Recognition*, 2015.

[6] X. Chen and L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[7] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[8] B. Dai, D. Lin, R. Urtasun, and S. Fidler. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[9] M. Daniluk, T. RocktÃd'schel, J. Welbl, and S. Riedel. Frustratingly short attention spans in neural language modeling. In *Proceed-ings of International Conference on Learning Repre-sentations, (ICLR)*, 2017.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Computation and Language*, 2016.

[11] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. DollÃąr, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From captions to visual concepts and back. In *InCVPR*, 2015.

[12] Y. Feng, L. Ma, W. Liu, and J. Luo. Unsupervised image captioning. In *CVPR*, 2019.

[13] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[14] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired training data. 2016.

[15] L. Huang, W. Wang, J. Chen, and X.-Y. Wei. Attention on attention for image captioning. In *ICCV*, 2019.

[16] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. Aligning where to see and what to tell: image caption with region-based attention and scene factorization. 2015.

[17] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, 2014.

[18] M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, and E. Erdem. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of*

the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Valencia, Spain, 2016.

[19] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[20] R. Kiros, R. Salakhutdinov, and R. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *Workshop on Neural Information Processing Systems*, 2014.

[21] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *Computation and Language*, 2014.

[22] I. Laina, C. Rupprecht, and N. Navab. Towards unsupervised image captioning with shared multimodal embeddings. In *International Conference on Computer Vision*, 2019.

[23] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out, pages 74-81, Barcelona, Spain, July. Association for Computational Linguistics.*, 2004.

[24] C. Liu, J. Mao, F. Sha, and A. L. Yuille. Attention correctness in neural image captioning. In *AAAI*, 2017.

[25] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention viaa visual sentinel for image captioning. In *CVPR*, 2017.

[26] J. Lu, J. Yang, D. Batra, and D. Parikh. Neural baby talk. In *CVPR*, 2018.

[27] R. Luo and G. Shakhnarovich. Controlling length in image captioning. In *Computer Vision and Pattern Recognition*, 2020.

[28] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks. In *International Conference on Learning Representations*, 2015.

[29] Y. Pan, T. Yao, Y. Li, and T. Mei. X-linear attention networks for image captioning. In *Computer Vision and Pattern Recognition*, 2020.

[30] D. Paperno, G. Kruszewski, A. Lazaridou, Q. N. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. FernÃąndez. The lambada dataset:word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the As-sociation for Computational Linguistics (Volume 1:Long Papers), pages 1525âĂŞ1534, Berlin, Germany.Association for Computational Linguistics*, 2016.

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[32] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek. Areas of attention for image captioning. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[33] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences forricher image-to-sentence models. In *Computer Vision and Pattern Recognition*, 2016.

[34] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li. Deep reinforcement learning-based image. captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[35] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Å. Kaiser, and I. Polosukhin. Attention is all you need. In *Conference on Neural Information Processing System*, 2017.

[37] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Computer Vision and Pattern Recognition*, 2015.

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[39] Y. Wang, Z. Lin, X. Shen, S. Cohen, and W. Cottrell, Garrison. Skeleton key: Image captioning by skeleton-attribute decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[40] Y. Wu, L. Zhu, L. Jiang, and Y. Yang. Decoupled novel object captioner. In *ACM MultiMedia*, 2018.

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2015.

[42] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen. Encode, review, and decode: Reviewer module for caption generation. In *30th Conference on Neural Image Processing System*, 2016.

[43] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *IEEE International Conference on Computer Vision*, 2017.

[44] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[45] L. Zhang, F. Sung, F. Liu, T. Xiang, S. Gong, Y. Yang, and T. M. Hospedales. Actor-critic sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

APPENDIX A

NETWORK DETAILS

Many different algorithms exist in the literature for the image caption task. However, the basic ideas of the networks in the algorithms are similar. In this section, details of the feature extracting networks will be explained.

A.1 Convolution Neural Network

CNN is actually a feed-forward neural network. However, instead of all neurons are communicated to the next layer's neurons, neurons are only associated with their neighbors. This spatial communication helps the network recognize the same pattern on the input without taking into account the pattern's location. Thus, this network structure is beneficial and famous for the computer vision area.

To compute neighbors of the pixel relationship, a filter (kernel) across all elements of the input as shown in Figure A.1

In the graph, the filter represents the weights of the feed forward neural network. The size of the filter is decided while designing the network. However, it is usually chosen smaller than the input's size to detect the small patterns on the input. Besides, CNN also shares the parameter of the filter while crossing the whole image. Thus, it utilizes less computation time and parameters comparing to the fully connected layer.

Convolution is also very similar to cross-correlation. The experiments show that cross-correlation and convolution provide the same performance. However, coding cross-correlation is much easier than convolution while dealing with backpropaga-

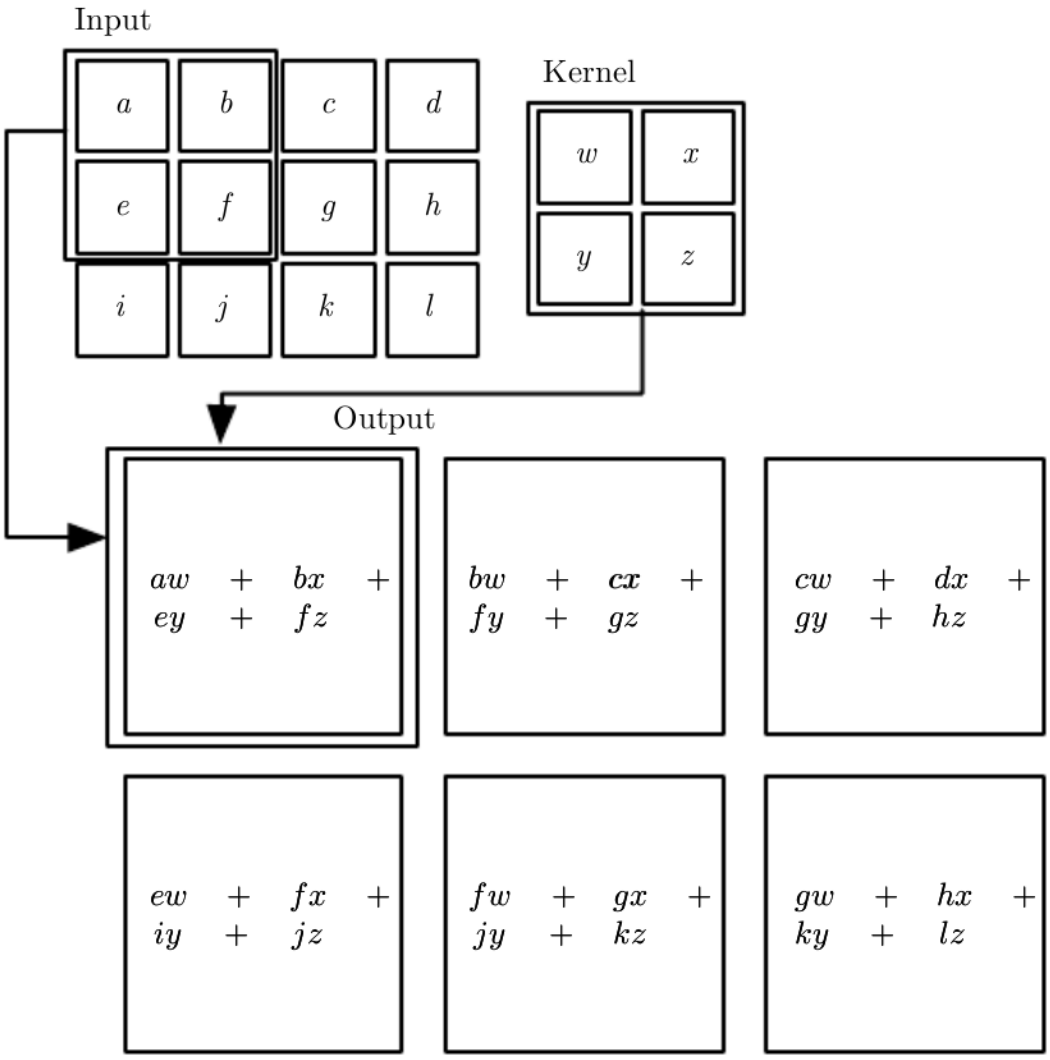

Figure A.1: Convolution Neural Network

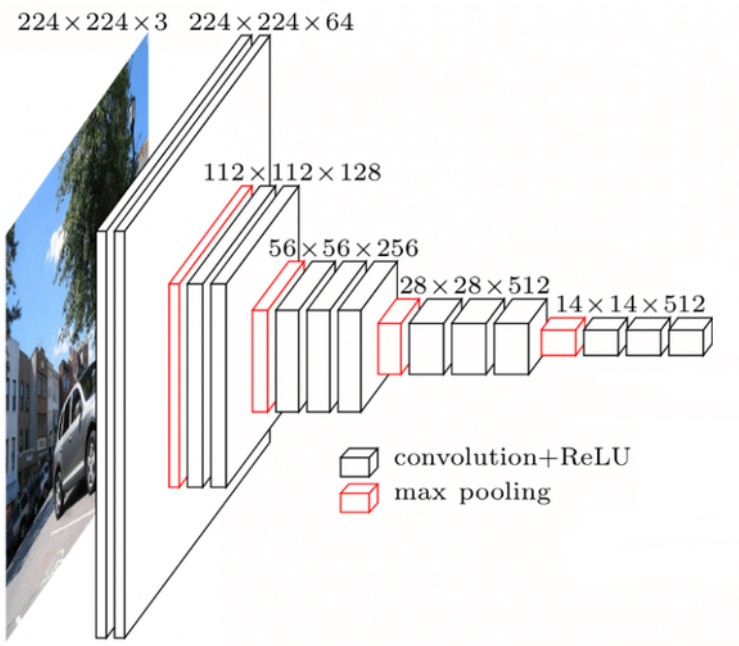

Figure A.2: Lenet Architecture

tion. Thus, cross-correlation is used instead of convolution. However, the network is called a convolution neural network in the literature.

Moreover, CNN is utilized for encoding abstract information of an image in the image caption task. One filter is not enough to accomplish this purpose. Thus, a more complex network such as Resnet, LeNet, or VGG-16 can be utilized. These networks were initially designed to detect objects on an image. However, the papers in the literature show that they are also very useful to understand the image's content in the image caption task.

A.2 Recurrent Neural Network

The sequence data is needed to process tasks such as activity recognition on video or natural processing language. However, since the feed-forward neural network focuses on the location of the input, one needs to train its feed-forward network for different combinations of the input for a robust model. This means too much data and training time. Therefore, RNN is specially designed to process the data sequentially. In contrast to the feed-forward network, vanilla RNN calculates a hidden state that encodes previous input information and represents the output of the RNN. The vanilla RNN

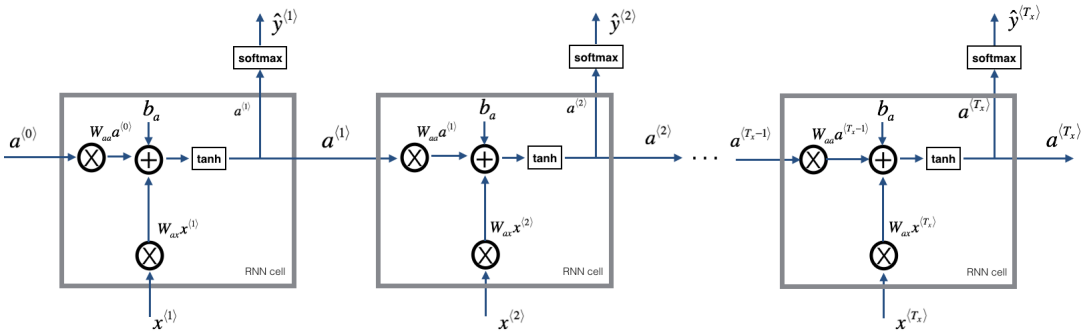

Figure A.3: RNN Architecture

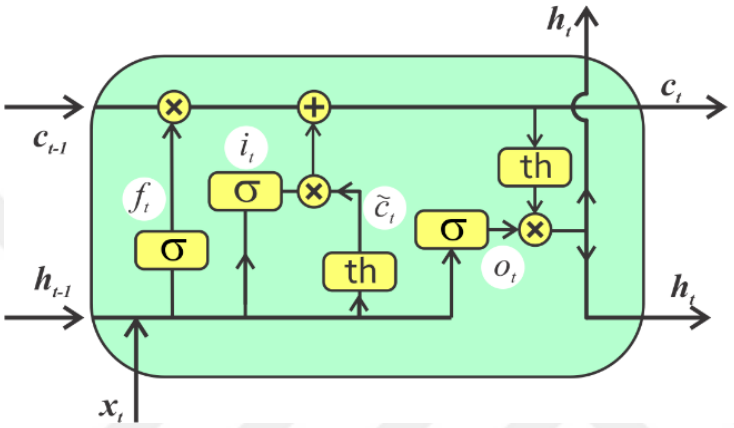

Figure A.4: LSTM Architecture

can be seen as below:

Since it uses the same weights for calculating next hidden state, it also shares its parameters.

Similar to CNN, RNN is also defined as feed forward neural network.

A.3 Long-short Term Memory

Even though RNN can manage the sequential data, it also has some drawbacks. Hidden state is calculated as follow $h_t = tanh(b + W_h^t * h_{t-1} + W_x * x_{t-1})$. Therefore, if values of W_h are greater than 1, $W_h^t * h_{t-1}$ will explode after one point. On the other hand, if values of W_h are smaller than 1, $W_h^t * h_{t-1}$ will vanish after one point. Thus, in either way, Rnn fails to convey information about the previous hidden state and input in further steps.

LSTM keeps information of previous input and hidden state in memory state instead of the hidden state to avoid this problem. Therefore, the hidden state only represents the output state of the LSTM. To calculate memory, how much information should be passed from the previous block and how much information should be added to the memory from current input are calculated using forget and input gates.

A.4 Transformer

Transformer [36] is a special network that tries to reduce the disadvantage of LSTM. LSTM calculates what to forget and what to remember information in the previous memory cell. Information in the memory is calculated recursively from previous states. Therefore, there is a risk of forgetting previous important information. Rather than determining what to remember or forget, transformer utilizes a multi-head attention mechanism to link words. Moreover, since transformer cannot use any recurrent network, it can be trained in a parallel manner. Position embedding vector is added to word embedding to understand the location of the word. The position vector can be calculated as follow:

$$w_{pe}^{n} = sin(pos * freq_k) \; if \; n = 2k \tag{A.1}$$

$$w_{pe}^{n} = cos(pos * freq_k) \; if \; n = 2k + 1 \tag{A.2}$$

$$freq_k = 1/(1000^{2k/dimension}) \tag{A.3}$$

After that, the relationship between words (object features for image caption tasks) is created using the transformer's encoder part. The multi-head mechanism is used to obtained that relationship information. Also, query, key, and value vectors are defined for the multi-head mechanism. Query vector (Q) refers to an abstract representation of where all information convey. Key vector (K) represents what we are looking for, and value vector (U) refers to the vector which passes information to the next layer. The scale dot attention mechanism computes a similarity score between query and

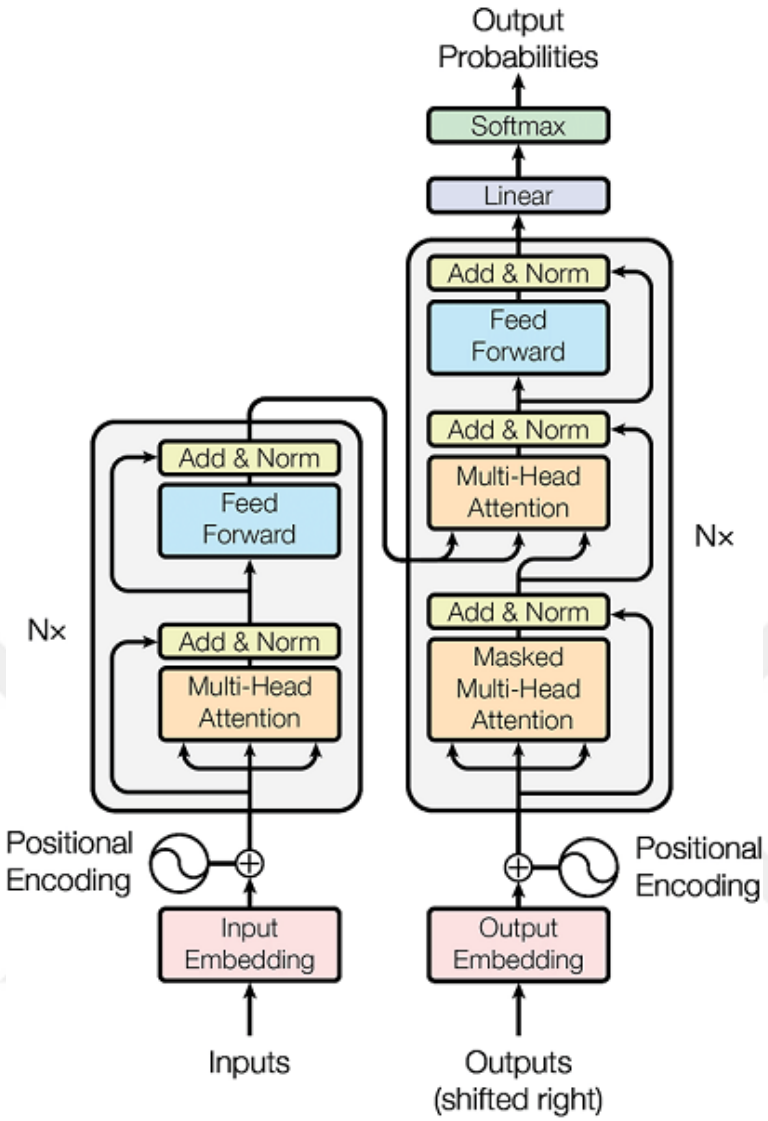

Figure A.5: Transformer Architecture

key vectors as an alignment score. The formula of scale dot attention as below;

$$Attention(Q, K, U) = softmax(\frac{QK^T}{\sqrt{D_k}})U \tag{A.4}$$

After the feed-forward layer, key and value vectors are sent to the decoder multi-head attention module. Query vector, on the other hand, is computed in the decoder part based on previous words. The architecture of the model can be found in Figure A.5. For the whole scene image caption task, the input vector is object features, and the outputs vector is previously generated words.