Image Caption Generator

Jashia Mitayeegiri
University of North Texas
Denton, Texas
Jashia..jm@gmail.com

Rahul Siddartha Gotti
University of North Texas
Denton, Texas
rahulsiddarthagotti@my.unt.edu

Abstract

The image captioning involves recognizing the context and annotating the given image by performing extraction of features from the image. It is a very eclectic task as it involves image processing and text processing. The techniques for handling these two different types of data include convolutional neural networks and recurrent neural networks which are used in a top down approach. The data set includes the images and the captions from the Flickr 8k data set and is followed by preprocessing of text. The ImageNet model VGG-16 issued to extract the features of the image and transform them into a vector and the LSTM are used handle the text sequence of those images. The dropout mechanism is used to avoid over-fitting and improve generalization for the LSTM. The decoder is used process the results from these layers to generate an apt caption to the image. The bilingual evaluation understudy -1 (BLEU -1), BLEU -2, and ROUGE are used to evaluate the model and to know the quality of the captions. The Word2vec model is used to create caption vector which is formed by taking the average of the word vectors and is used to measure the image similarity. The ablation study is conducted to analyze the contribution of components.

1. Introduction

Please follow the steps outlined below when submitting in every walk of our life we perceive objects, events and environment around us. We can recognize objects within no time and respond to the environment and label them as quickly as possible and moreover they we receive no description of them. It is complex how the brain works regarding such generation

The main challenge with the image captioning task is to design a model which could fully utilize the image provided and extract a rich description like a human would describe the image. Object detection is not enough to generate the caption, the relationship and the attributes of them is also needed. On top of analyzing the objects and its relationships, we need to come up with a meaningful description of image. This process also requires the thorough understanding of the world and the impact of every minimalistic detail.

The advent of CNN in computer vision and RNN in the natural language processing has been a breakthrough for both fields. The CNN are heavily used because of its capability to extract prominent features from the image which makes is suitable for image classification and object detection. The RNN give prime importance to the sequence and hence this ability of it makes it suitable for language modeling and machine translation. The combination of these both methods will enable us to generate captions to images.

The model built can be widely used for the various applications like allowing the visually impaired human being and to generate data sets with the labels which saves the time, effort and skip the mundane work of labelling the images. As it is the era of social media, generating image descriptions can enable the reachability and accessibility to the images. Another application is E-commerce, where the detailed description of products can help the customers to understand it better. It can also be used in surveillance cameras to describe the event happening in real time.

The captions generated can be further used to find the similarity between the images. This will benefit various businesses like E-Commerce which will enable the customer to find relevant products. In the healthcare domain this feature will be a useful aid to diagnose and prepare the treatment plan or monitor progress. It can be used in entertainment domain to give out the movies names by finding the similarity between movie posters.

2. Related Work

The [1] discusses the approach based on the bag-of-words where each word in the caption is a separate variable. The features of the image are extracted using the pre-trained object recognition model, maximum entropy classifier is used for training the model which enable it to

learn about the distribution of words with respect to the features. The Ryan Kiros et al [3] aims to establish a visual semantic embedding model which maps the images and the test description in a common feature space where the similar feature vectors are close together.

The A.Aker et al [2] used the dependency relation between the objects in the image and their attributes and showed a improved performance for the image in generating the image captions. The approach of extracting the features from image and text using the Latent Dirichlet Allocation is done by Ragkhitwetsagul et al [7] where the image and text are represented as latent topics which are later used to generate the captions. The paper [4] proposes an end to end dense approach where the CNNs are used for object detection to generate object proposals and then RNN is used to generate captions for each object proposal.

The [8] introduced an innovative approach for caption generation of images, called as "Show and Tell" and proposed the combination of CNN and RNN by considering the objectives of image caption generation. The main idea is to use CNN to encode the image into a vector which is used to initiate the hidden state of RNN and capture the global context and content of the image.

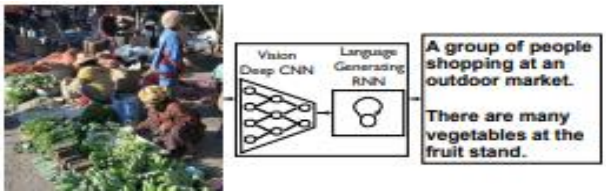

Figure 1: Show and Tell mechanism.

The limitation of the above approach is that the part of the image which is a background and has no role to play has been given equal weight along with important parts of the image. The [9] redefines it by calling "Show, Attend and Tell" where the attention is given to the feature which contribute for making better captions. The attention weights are computed based on the current hidden state of RNN and the features extracted.

The [5] was a bottom-up approach which used the multimodal object function which is used to align the visual and semantic representation of words. This approach enables us to enable the mapping of image to text with a consistent meaning. The paper [6] proposes the use of a single neural network which is jointly optimized to predict right captions and is trained using the maximum likelihood estimation which maximizes the likelihood of the correct caption to the given image.

Topic modelling has been an effective approach to understand huge corpus of text. The paper [10] introduces the latent semantic analysis (LSA) which finds the latent semantic patterns in the text. The paper maps the words and documents in the corpus to a lower dimensional space using singular value decomposition (SVD) as it helps in reducing the dimensions of the data.

The paper [11] induces the probability to LSA where it generates topic-word probability and document topic probability. These probabilities are estimated using the maximum likelihood estimation. The Laurens et al[12] presents the nonlinear dimensionality reduction technique called as t-distributed stochastic neighbor embedding (t-sne) which makes sures that the similar data points lie close to each other even after the dimensional space is reduced and is used for visualization of data.

The [13] paper is significant in the field of natural language processing as it introduces the idea of word2vec model and the two variants of it called as continuous bag of words (CBOW) and skip gram model. This model learns the syntactic and semantics between the context word and its neighbors. The word2vec has been state of the art model which is used to generate the effective word vectors. The [14] builds up on the word2vec model which extends the architecture of it to embeddings of phrases and multi word expressions.

Pennington et al [15] introduces an approach called Global Vectors (Glove) which learns the word embeddings using the matrix factorization. This method differs from the word2vec model as it uses the factorization of global word-word co-occurrences matrix whereas the word2vec uses the context of the word to predict the word.

3. Proposed Approach

The idea of image caption generator is to analyze the image and make the semantically correct caption to the image. The CNN are extremely efficient when dealing with images and extract the features passing through different filters/kernels so that these features can be used to caption the image. The RNN are specifically used to generate sequences and these networks can be used to generate the sequence of words which are syntactically and semantically right. The overview of the architecture is as in Figure 3.

we use the pre-trained models from the ImageNet competition, these are very deep CNN models which are trained on the ImageNet database. The model we use is VGG-16, the weights of this model are frozen, and these weights are used to extract the features in the image by removing the classification layer and taking the output of 2nd fully connected. This is an efficient way because no new model needs to be created to learn the features of the

image and the pre-trained models can be used producing efficient results. The VGG-16 consists of 13 convolutional layers and 3 fully connected layers. The convolutional layers are 3 X 3 convolutional layers with a stride size of 1 and the pooling layers are all 2 X 2 pooling layers with a stride size of 2. the Figure 1 shows the architecture of it. The VGG -19 model is also used in place of VGG-16 which is much deeper than VGG – 16.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2: VGG-16 Architecture

The Long Short-Term Memory (LSTM) are a specific type in RNN which are designed to handle the vanishing gradient problem, encountered during the training process. This makes it hard for the RNN to predict the next word sequence as it forgets the context of the event. In image captioning generation the LSTM based models are well suited as they are capable of handling various length of inputs and produce output by capturing the complex interactions between the features and the text. The Figure 2 shows the cell of LSTM where the forget gate enables the LSTM to discard the previous context and obtain the current context of the event. This context is passed on, so that the vanishing gradients problems do not arise. The memory cell of LSTM consists of three types of gates which control the context to be passes and the gates are as follows:

1. Input gate: It determines the amount of new information to be stored in memory cell.
2. Forget gate: It determines the amount of memory to be kept in the memory cell.
3. Output gate: It determines the amount of current memory cell state to be used in the final hidden state output.

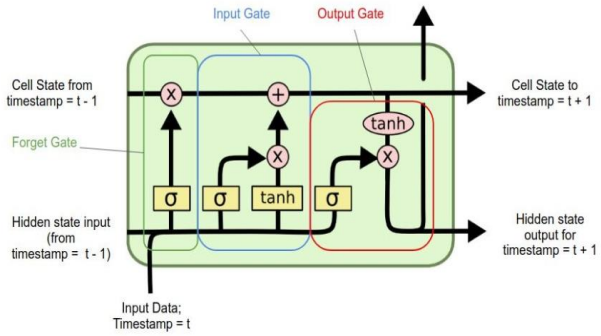

Figure 3: LSTM Memory Cell

The Gated Recurrent Units (GRU) are also used in place of LSTM as they are computationally less expensive and are faster than LSTM. It has fewer parameters and a simplified structure which makes it memory efficient too. The GRUs are also used to provide a solution to the vanishing gradient problem and have a memory cell which is built using the following gates:

1. Reset gate: It controls the amount of information to be forgotten from the previous hidden state.
2. Update gate: It controls the amount of information to be kept from the current hidden state.

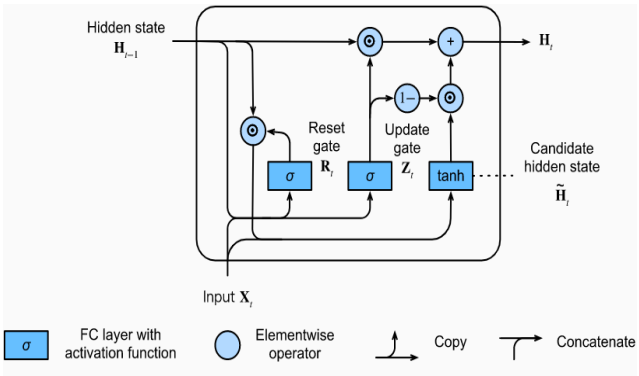

Figure 4: GRU Memory Cell

When the captions are generated, those captions are used to find the similarity between the images and the word2vec model is used. The word2vec model is used to generate the word embedding for a given word. It is a pretrained model and uses its weights to generate the vectors for words. The figure below is the architecture of word2vec model, and the hidden layer weights are used to produce word embeddings. From this word embedding the caption vector are generated to find the similarity.

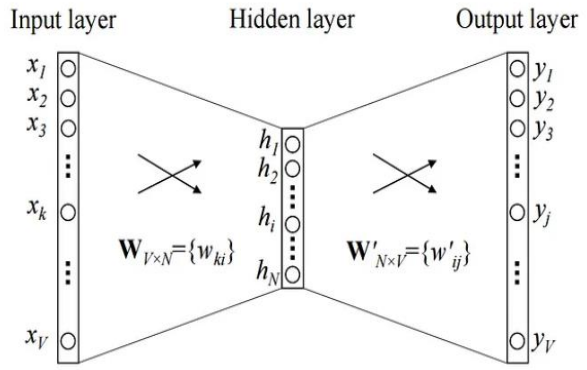

Figure 5: Word2vec Architecture

The model built must take two types of inputs which include the images and their captions. The VGG-16 model and the LSTM/GRU are used to as encoder model and are merged to produce input to the decoder part which predicts the words based on the given input. The component of the model is as follows.

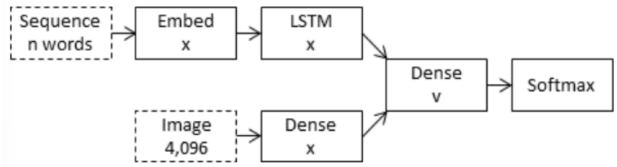

Figure 6: Component diagram of the model

4. Implementation

The first phase is knowing about the data. As the images are described using captions, we performed topic modeling using those captions so that we know the major topics that are involved in the dataset and this helps to understand the flickr8k dataset better. For the implementation of topic modeling, we have used the latent semantic analysis function from genism library which outputs the topics and the words to describes those topics. These topics are plotted using the t-sne.

```
[(0,
  '0.769*"dog" + 0.258*"black" + 0.249*"white" + 0.217*"two" + 0.196*"brown" + 0.171*"man"'),
 (1,
  '0.692*"man" + -0.372*"dog" + 0.226*"woman" + 0.213*"wearing" + 0.188*"boy" + 0.187*"girl"'),
 (2,
  '0.555*"two" + -0.537*"man" + 0.356*"girl" + 0.328*"boy" + 0.185*"young" + -0.141*"dog"'),
 (3,
  '0.634*"two" + -0.384*"boy" + -0.292*"girl" + -0.229*"white" + 0.222*"man" + -0.184*"black"'),
 (4,
  '0.698*"woman" + -0.321*"man" + -0.298*"boy" + 0.261*"white" + 0.231*"black" + 0.220*"wearing"'),
 (5,
  '-0.759*"girl" + 0.582*"boy" + -0.100*"little" + 0.090*"two" + 0.083*"blue" + 0.083*"shirt"'),
 (6,
  '-0.567*"woman" + 0.508*"white" + 0.355*"people" + 0.237*"black" + -0.231*"boy" + -0.209*"dog"')]
```

Figure 7: Topics generated by topic modeling.

The first phase also involves pre-processing of text which involves cleaning the data to improve the quality of data. During this phase <start> and <end> are used as identifiers for each caption which are used as indicators for the LSTM model. Tokenization is also used to get the unique list of words and assign the indexes to every word in the corpus. The vector of each caption is generated by replacing the indexes with the words.

```
mapping['1000268201_693b08cb0e']

['startseq child in pink dress is climbing up set of stairs in an entry way end
seq',
 'startseq girl going into wooden building endseq',
 'startseq little girl climbing into wooden playhouse endseq',
 'startseq little girl climbing the stairs to her playhouse endseq',
 'startseq little girl in pink dress going into wooden cabin endseq']
```

```
seq = tokenizer.texts_to_sequences(mapping['1000268201_693b08cb0e'])[0]
print(mapping['1000268201_693b08cb0e'][0])
print(seq)
```

```
startseq child in pink dress is climbing up set of stairs in an entry way endseq
[1, 42, 3, 88, 172, 6, 115, 50, 388, 11, 384, 3, 27, 5013, 669, 2]
```

Figure 8: Caption to vector

The second phase is generating the feature vector of the images which uses the pretrained model called VGG-16. In the VGG-16 model the classification layer is removed and the output of 2nd fully connected layer which is (1,4096) vector for each image is used to feed to the model. The VGG -16 model has the following parameters after the removal of the classification layer.

```
block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0
flatten (Flatten)           (None, 25088)             0
fc1 (Dense)                 (None, 4096)              102764544
fc2 (Dense)                 (None, 4096)              16781312
=================================================================
Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0
_________________________________________________________________
```

Figure 9: Output layers of VGG-16

Figure 10: Image to Vector

The final phase is to build the model which consists of an encoder and decoder model. The encoder model takes two types of inputs which are sequence of indexed captions and the feature vector of the image. The sequence vector is of 1 X 31 dimension as the longest captions are of 31 words long and if the current caption consists of shorter size then it padded to make it to 1 X 31 dimension. This is later passed to the embedding layer which converts the integer indexes to a dense vector of dimension 1 X 31 X 256 followed by a dropout layer with the dropout rate of 0.4. At last, the LSTM or GRU units are added to perform the RNN process.

The second input to the model is a feature vector of 1 X 4096 vector which is passed through a dropout layer which has dropout rate of 0.5 and is later passed to a fully connected which makes to vector of 1 X 4096 to 1 X 256 dimensions which is equal to the final dimension of the sequence input. These two outputs are merged and passed on to the decoder which has a fully connected layer with 256 units and relu as activation function. This is given as an input to the last fully connected layer which as SoftMax as an activation function. The final layer produces a vector of size 1 X 8313 as the vocabulary of the dataset is 8313, the SoftMax produces the probability of each word and then word with the maximum probability is given as an output.

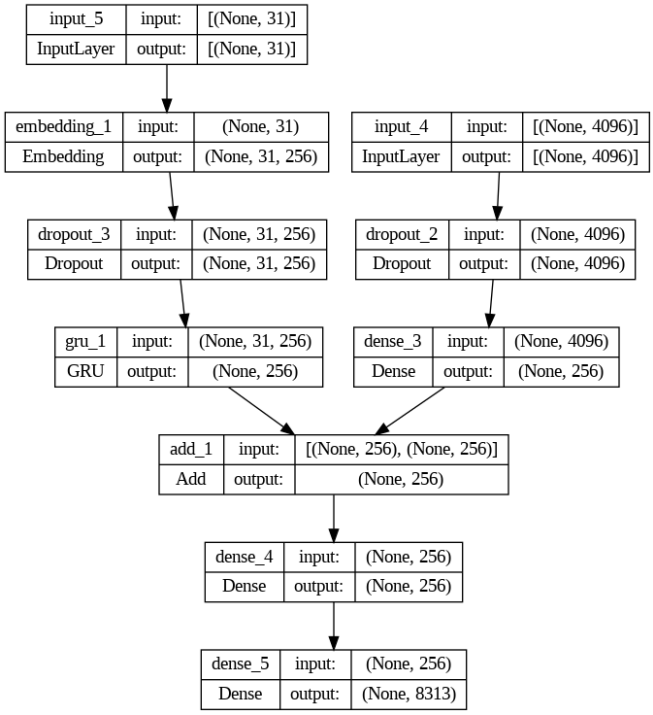

Figure 11: The architecture of the model built.

The captions generated are tokenized and each token is given a vector by the word2vec model. The average of these word vectors will enable us to produce a vector for a caption. Cosine similarity is used to measure the similarity between the two captions which represent the image. The word2vec model is first fitted with the corpus dataset with a minimum count of 1 and with the sliding window of 2 is used.

The model is trained with batch size as 32 and with 25 epochs where the steps per epoch is the ratio of length of training data and the batch size which is 227. The classifier used is ADAM with a learning rate of 0.001, the loss is calculated using the categorical Cross entropy.

```
Epoch 21/25
227/227 [==============================] -
716s 3s/step - loss: 2.0711
Epoch 22/25
227/227 [==============================] -
725s 3s/step - loss: 2.0469
Epoch 23/25
227/227 [==============================] -
725s 3s/step - loss: 2.0225
Epoch 24/25
227/227 [==============================] -
699s 3s/step - loss: 2.0014
Epoch 25/25
227/227 [==============================] -
678s 3s/step - loss: 1.9815
```

Figure 12: Training of the model

5. Experimental analysis and Results

5.1. Results

The dataset is Flickr8k dataset which consists of 8000 images and a captions text file. Each image in the data set has its respective caption where the image is described using five different captions which provide a clear description of the image. This description is concise providing the salient features, events, and context of the image. The images and the captions.txt file are shown in Figure 5 and Figure 6.

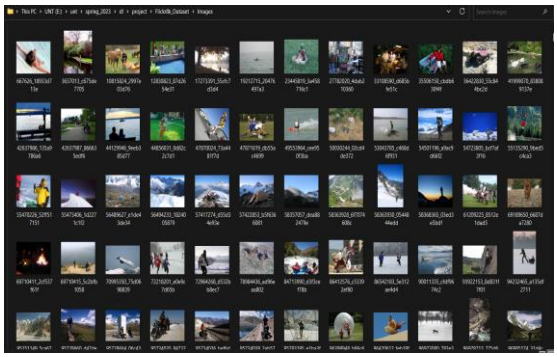

Figure 13: Flickr 8k images

captions

File Edit View

```
image,caption
1000268201_693b08cb0e.jpg,A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg,A girl going into a wooden building .
1000268201_693b08cb0e.jpg,A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg,A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg,A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg,A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg,A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg,A black dog and a white dog with brown spots are staring at each other in the street
1001773457_577c3a7d70.jpg,Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg,Two dogs on pavement moving toward each other .
```

Figure 14: Flickr 8k captions.txt file

Figure 15: LSTM forget gate

The description of the Figure 7 in the captions.txt file is as follows:

1. A mother and children is fishing on a boardwalk at night.
2. woman and three children stand on a deck with a fishing pole .
3. A woman stands with children on a boardwalk at night overlooking the sea .
4. Some people on a pier at night with one girl fishing off
5. Woman with three children fishing over boardwalk in the evening .

5.2. Results

To have a brief idea about the dataset we have applied topic modelling and have plotted the prominent topics using t-sne.

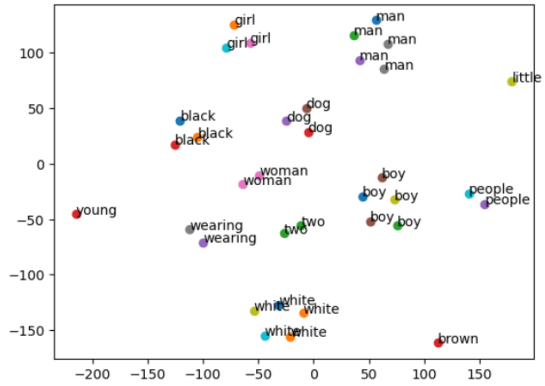

Figure 16: Topics plotted using t-sne

The generated caption for the image by the LSTM is as follows:

```
---------------------Actual---------------------
startseq blonde child swinging on swing endseq
startseq smiling child wearing white t-shirt with stripes and skull and crossbo
nes is swinging endseq
startseq young child in swing wearing skull and crossbones shirt endseq
startseq the blonde haired child played on the swing endseq
startseq the young girl in cartoon shirt is enjoying ride on swing endseq

--------------------Predicted--------------------

'startseq blonde toddler in pink shirt and blond hair sitting on tire swing end
seq'
```

Figure 17: Output of GRU model

The generated caption for the image by the GRU is as follows:

```
--------------------Actual--------------------
startseq blonde child swinging on swing endseq
startseq smiling child wearing white t-shirt with stripes and skull and crossbo
nes is swinging endseq
startseq young child in swing wearing skull and crossbones shirt endseq
startseq the blonde haired child played on the swing endseq
startseq the young girl in cartoon shirt is enjoying ride on swing endseq

--------------------Predicted--------------------

'startseq little girl is laughing on swing endseq'
```

Figure 18: Predicted output of LSTM model

The similarity of the images is found using the cosine similarity between the caption vectors of two images.

Table 1: Image similarity scores

Captions and Images	
'startseq bunch of people in life jackets and jackets on both orange boats endseq' 	
'startseq two boys are sitting in the water with two boys in the background endseq' 	0.7839
'startseq dog is runnin gthrough the grass endseq' 	0.3670

5.3. Evaluation of model

5.3.1. BLEU (Bilingual Evaluation Understudy)

The BLEU is a primarily used to measure the quality of text generated by the model. It measures the similarity between the generated caption and the reference captions using the frequency of subsequence of generated captions in the reference captions. The precision of each subsequence is taken and the geometric mean gives the BLEU score. It's one of the advantages is brevity penalty which makes sure that the longer captions are generated than the short ones.

BLEU focuses much on the precision and looks how much overlap does the generated captions have with respect to the reference captions. Its value ranges from 0 to 1, the higher the score the better is the performance, the n in n-grams can also be used to fix the n-grams. For our project we have used BLEU-1, BLEU-2 and BLUE -3 so we will be using 1-grams, 2-grams and 3-grams.

$$Geometric\ Average\ Precision\ (N) = exp(\sum_{n=1}^{N} w_n\ log\ p_n)$$

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

Bleu (N) = Brevity Penalty X Geometric Average Precision Score (N)

Table 2: BLEU scores

	LSTM	GRU
BLEU - 1	0.4640	0.5252
BLEU - 2	0.2252	0.3031
BLEU - 3	0.0012	0.0034

5.3.2. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

The ROUGE score is used for summarization and machine translation, so having this as one of the metrics will enable us to focus on how much of the reference captions was captured by generated captions. There are mainly three types of ROUGE score:

- ROUGE – 1 – calculates the overlap between unigrams of generated and reference captions.
- ROUGE- 2 - calculates the overlap between bigrams of generated and reference captions.
- ROUGE-L - calculates the longest common sequence (LCS) between generated and reference captions.

Table 3: ROUGE Score of LSTM

	Recall	Precision	F-Score
Rouge - 1	0.1808	0.4951	0.2605
Rouge - 2	0.0222	0.0912	0.0356
Rouge - L	0.1672	0.4605	0.2412

Table 4: ROUGE Score of GRU

	Recall	Precision	F-Score
Rouge - 1	0.1928	0.5592	0.2822
Rouge - 2	0.0338	0.1462	0.0522
Rouge - L	0.1787	0.5202	0.2618

5.4. Ablation Study

The ablation study is used to study the contribution of the model components. During this process one or more components are removed systematically to see how it affects the model performance. This will also enable us to replace the components which might perform better than the existing once or remove the component completely if it does not degrade the model performance or improve the model performance.

The pretrained ImageNet models are used to get the image feature vector, we have replaced the VGG-16 model with VGG-19 model which is deeper than the VGG-16 and see how the model performance. The loss of the model using VGG-16 and VGG -19 is as follows:

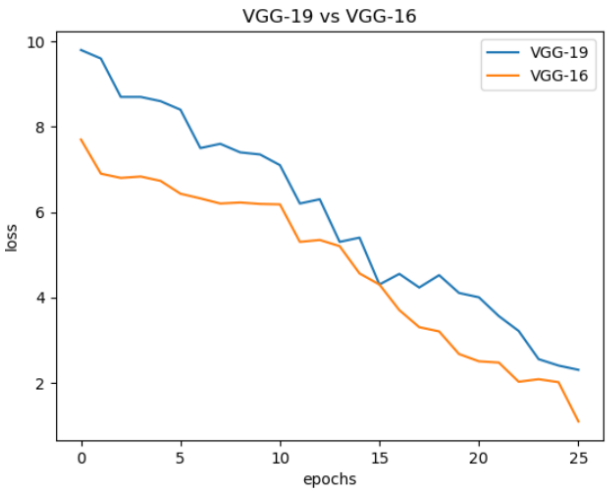

Figure 19: Performance of the model VGG-16 vs VGG-17

As VGG-16 low loss when we compare it with the VGG-19, we will be using the VGG-16 model to extract the feature vectors of the images.

We do have options in VGG-16, the VGG-16 has two fully connected at the end which are of dimension 1 X 4096. We have used the 2nd fully connected layer output and generated the feature vectors for each image. Now we use the 1st fully connected layer output to generate the image feature and monitor the results as well as the 2nd . We see that the 1st fully connected layer has increased loss when we compare it with the 2nd fully connected layer. So we will be using 2nd fully connected layer.

Table 5: Model performace with FC1 and FC2

	1st Fully Connected Layer (VGG-16)	2nd Fully Connected Layer (VGG-16)
LSTM	3.6780	2.0554
GRU	3.3021	1.9870

The LSTM are replaced with GRU as both are used for vanishing gradient problem and helps to preserve the long-term dependencies. The loss of the model is plotted, and we see that the GRUs outperform the LSTMs.

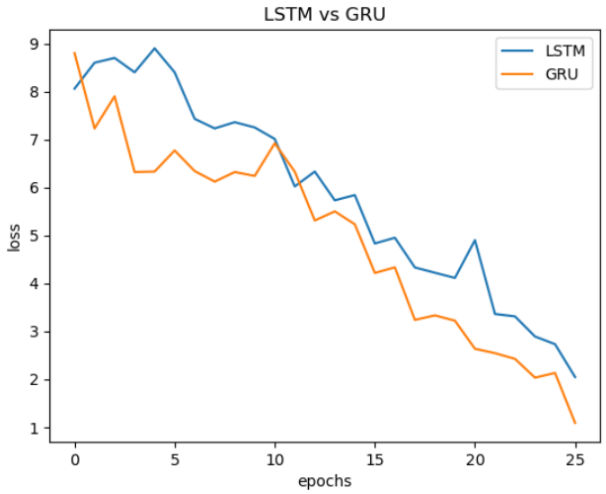

Figure 20: Model Performance with LSTM and GRU

6. Conclusion

Image caption generation is an eclectic task as it involves two major fields which include computer vision and natural language processing. The model built for this task is a merge model architecture as it takes two inputs from the encoder and merges it to pass on to the decoder. The VGG-16 has performed better than VGG-19. Fine tuning, dropout mechanism and replacing the LSTM with GRU has significantly improved models' performance. The better the model generates the captions the better the word2vec could allow us to calculate the similarity between the images effectively. The use of large dataset like Flickr 30k and addition of attention would push the model further better.

References

[1] Farhadi, Ali, et al. "Every picture tells a story: Generating sentences from images." Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11. Springer Berlin Heidelberg, 2010.

[2] Aker, Ahmet, and Robert Gaizauskas. "Generating image descriptions using dependency relational patterns." Proceedings of the 48th annual meeting of the association for computational linguistics. 2010

[3] Kiros, Ryan, Ruslan Salakhutdinov, and Richard S. Zemel. "Unifying visual-semantic embeddings with multimodal neural language models." arXiv preprint arXiv:1411.2539 (2014).

[4] Johnson, J., Karpathy, A., Fei-Fei, L. (2016). DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4565-4574).

[5] Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[6] Yang, Zhongliang, et al. "Image captioning with object detection and localization." Image and Graphics: 9th International Conference, ICIG 2017, Shanghai, China, September 13-15, 2017, Revised Selected Papers, Part II 9. Springer International Publishing, 2017.

[7] Ragkhitwetsagul, Chaiyong, Jens Krinke, and Bruno Marnette. "A picture is worth a thousand words: Code clone detection based on image similarity." 2018 IEEE 12th International workshop on software clones (IWSC). IEEE, 2018.

[8] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[9] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. PMLR, 2015.

[10] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407.

[11] Hofmann, T. (1999). Probabilistic latent semantic analysis. In Proceedings of the Seventeenth conference on Uncertainty in Artificial Intelligence (pp. 289-296). Morgan Kaufmann Publishers Inc.

[12] van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space." Proceedings of the International Conference on Learning Representations (ICLR), 2013.

[14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In Advances in Neural Information Processing Systems (pp. 3111-3119).

[15] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543.