Advanced Image Captioning

Parth Maheshwari
Michigan State University
mahesh22@msu.edu

Vishal Kumar Panda
Michigan State University
pandavis@msu.edu

I. Introduction

Recent advancements in deep learning techniques for perceptual tasks, such as image classification and object detection, have inspired researchers to address more challenging problems where recognition is merely a starting point for complex visual reasoning. Image captioning is one such task. The goal of image captioning is to automatically describe an image using one or more natural language sentences. This problem combines computer vision and natural language processing, and its primary challenges stem from the need to translate between two distinct yet often paired modalities. First, it is crucial to identify objects in the scene and determine their relationships, and then accurately convey the image content using well-structured sentences.

The generated descriptions, however, still differ significantly from how humans describe images, as people rely on common sense, experience, and focus on essential details while ignoring implied objects and relationships. Furthermore, humans often use imagination to create vivid and engaging descriptions. Despite these limitations, image captioning has already demonstrated its usefulness in applications such as assisting visually impaired individuals with daily tasks, content-based retrieval, and social media communications.

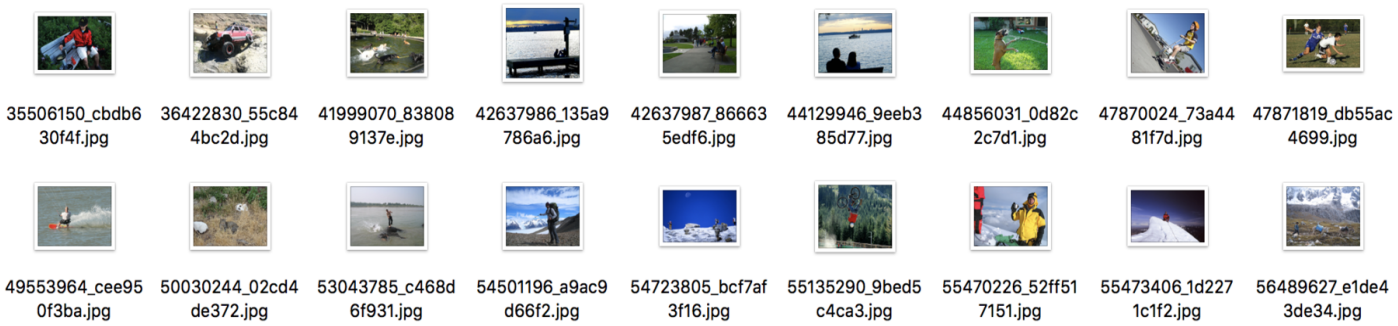
Some researchers have reformulated image captioning as a ranking task, where ranking-based approaches always produce well-formed sentences. However, these approaches cannot generate new sentences or describe compositionally new images, i.e., those with objects observed during training but appearing in different combinations in the test image. In contrast, today's state-of-the-art models are generative and based on neural networks. They typically employ an encoder-decoder architecture, combining a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN).

II. Problem Statement

Image captioning involves the automatic generation of one or more natural language sentences to describe an image. In recent years, the field has advanced rapidly, transitioning from initial template-based models to contemporary deep neural network-based approaches. This report provides a comprehensive overview of recent image captioning research, focusing specifically on models that employ the combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. We examine the merits and drawbacks of various strategies, as well as review prevalent evaluation metrics and datasets commonly used in the field.

A. Data Specification

- Flickr8k_Dataset: Contains a total of 8000 images in JPEG format with different shapes and sizes. Of which 6000 are used for training, 1000 for test and 1000 for development.
- Flickr8k_text: Contains text files describing train_set, test_set. Flickr8k.token.txt contains 5 captions for each image i.e. a total of 40460 captions.



Flickr 8k dataset

```
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 .
```

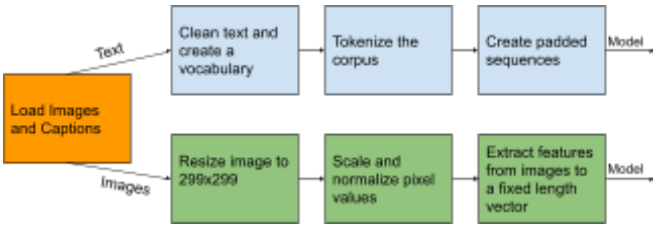
Sample Image and captions from the dataset

III. Methodology

In the introduction, we briefly overview the image captioning problem and its applications. In this section, we will go into the implementation details of our solution and the steps we followed to achieve high-quality image captioning. We used Python to perform these steps and Keras to implement our deep-learning models.

A. Preprocessing Pipeline

Vectorization of both text and image data formats was required to use them as model outputs and inputs respectively. The following basic steps were performed for the preparation of data across all our models.

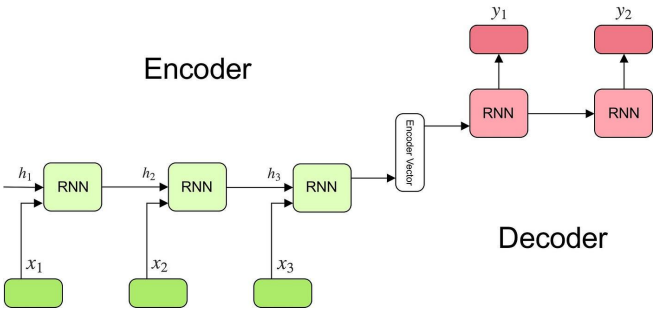


B. *Encoder-Decoder Architecture*

An encoder-decoder architecture is a type of neural network architecture used in natural language processing and other sequence-to-sequence learning tasks, such as machine translation, text summarization, and speech recognition.

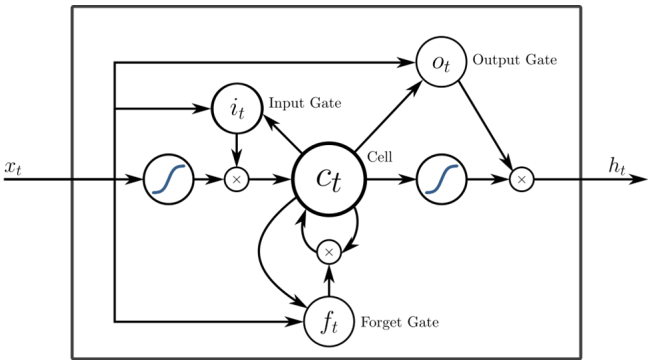
The encoder-decoder architecture consists of two main components: an encoder and a decoder. The encoder processes the input sequence (e.g., a sentence or a speech signal) and encodes it into a fixed-size vector representation, which captures the essential information of the input. The decoder then takes the encoded representation as input and generates the output sequence (e.g., a translation or a summary) one token at a time.

The encoder and decoder can be implemented using different types of neural networks, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), or transformer networks. In the case of RNNs, the encoder and decoder are typically implemented using long short-term memory (LSTM) or gated recurrent unit (GRU) cells.



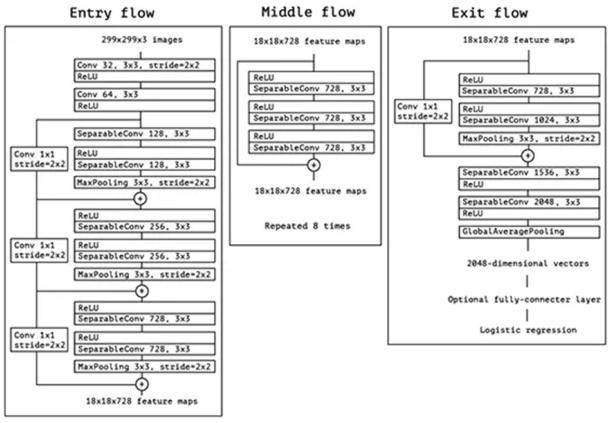
C. *Long Short Term Memory (LSTM)*

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) and were refined and popularized by many people in the following work. They work tremendously well on a large variety of problems and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of neural networks. LSTM is used to predict the sequence of words such as Image captions. It will be used as a decoder to generate captions.



D. *Depthwise Separable Convolutions - Xception*

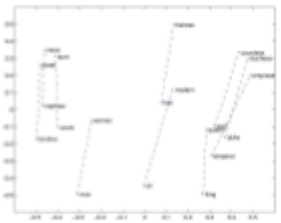
The Xception model is a pre-trained deep convolutional neural network architecture originally proposed by Google in 2016.split the convolutional operation into two separate steps: a depthwise convolution that applies a single filter to each input channel separately, followed by a pointwise convolution that applies a 1x1 filter to combine the results of the depthwise convolution. This approach reduces the number of parameters in the network and makes it more computationally efficient than other architectures.It is an extension of the inception Architecture which replaces the standard Inception modules with depthwise Separable Convolutions. It is used in Image captioning for feature extraction which will be used as an encoder to extract features from an Image



E. Pre-trained Word Embedding - GloVe

After training the basic model architecture discussed above in section 3.2, we found the sentences to be incoherent. While the generated captions correctly identified objects in the images, they were providing a wrong description of the overall scenario in the image. For eg, for the following image, our initial model without any pre-trained word embeddings generated the caption - “dog is running on the beach”

We can clearly see that the dog is swimming in a pool, not running on the beach



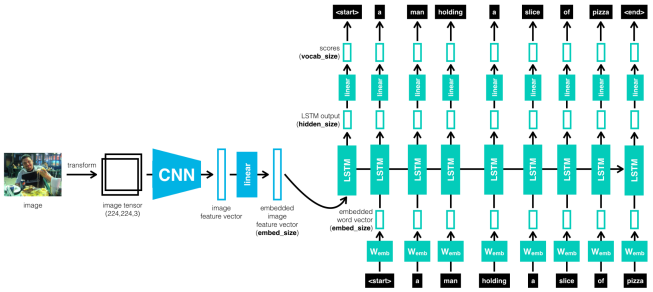
Word representation in GloVe

This was potentially a result of the model learning the word representation from scratch from very few (8k) captions, leading to a poor understanding of word context/representation. To overcome this issue, we used pre-trained GloVe (Global Vectors for Word Representation) to create word embeddings for the vocabulary in the captions. These embeddings will be used as inputs to the LSTM.

F. Architecture of the Model

To tackle the image-to-sequence problem of caption generation, we used the CNN-LSTM architecture, as it contains specialized layers to learn from both text and images. The CNN encoder essentially identifies patterns within images and converts them into a vector. The CNN encoder is followed by a recurrent neural network (LSTM) that generates a corresponding sentence. The feature vector is fed into the "DecoderRNN" (which is "unfolded" in time). Each word appearing as output at the top is fed back to the network as input (at the bottom) in a subsequent time step until the entire caption is generated. The arrow pointing to the right that connects the LSTM boxes together represents hidden state information, which represents the network's "memory", also fed back to the LSTM at each time step.

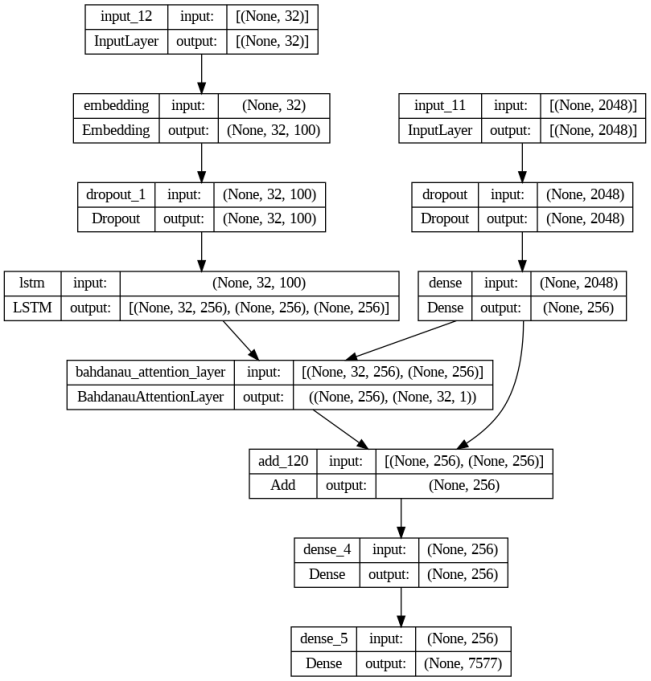
Once the <end> token is encountered, the complete caption is produced, providing a description of the given image.



The feature extraction model used is a pre-trained model Xception that given an image is able to extract the features often in the form of a fixed-length vector. It is trained on an imagenet dataset with 1000 different classes to classify the images. A deep convolutional neural network, or CNN, is used as the feature extraction submodel. This network is trained directly on the images in our dataset.

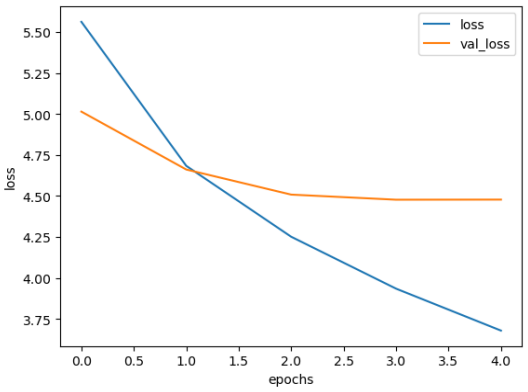
Keras Model has been used in order to define the structure of the model which includes:

- Feature Extractor –With a dense layer, it will extract the feature from the images of size 2048 and we will decrease the dimensions to 256 nodes.
- Sequence Processor – Followed by the LSTM layer, the textual input is handled by this embedded layer.
- Decoder – We will merge the output of the above two layers and process the dense layer to make the final prediction.




```
Total params: 5,134,490
```

Model Architecture (with attention+glove)



Loss Curve

IV. Results

A. *Evaluation Metrics*

We are using BLEU (Bilingual Evaluation Understudy Score) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics for evaluating generated captions to reference test captions.

BLEU works by counting matching n-grams in the candidate text to n-grams in the reference text, where 1-gram or unigram would be each token and a bigram comparison would be each word pair.

Whereas ROUGE, on the other hand, focuses on the recall of n-gram overlap between the generated summary and one or more reference summaries. ROUGE also considers different types of n-grams, including unigrams, bigrams, trigrams, and longer sequences of words.

B. *Train-Test split*

We performed a 75:25 train-test split for our experiments, resulting in over 6000 images and 30,000 captions in our training dataset. The evaluation metrics were calculated over 2000 test images and 10,000 test captions.

C. *Performance Results*

We evaluated our four models, namely -

A. Vanilla CNN-LSTM
B. CNN-LSTM with GloVe Embeddings
C. CNN-LSTM with Bahdanau Attention
D. CNN-LSTM with Bahdnau Attention and GloVe Embeddings

on multiple variants of our evaluation metrics, namely -

A. BLEU-2
B. BLEU-3
C. BLEU-4
D. ROUGE-precision
E. ROUGE-recall
F. ROUGE-f1

to get a comprehensive overview of how the models are performing on different length sequences (individual tokens, short sequences, and long sequences) in terms of verbosity (precision) and accurate word selection(recall).

The combined results across all experiments can be seen in **Table 1**. In the table, we observe that adding GloVe embeddings and Attention mechanisms to the model individually is resulting in a marginal increase in BLEU metrics compared to the baseline Vanilla CNN-LSTM model. Adding GloVe embeddings in fact worsened the model's performance compared to the baseline in terms of ROUGE scores. Since, the results seem contradictory, and BLEU scores seem very low in general, we will perform error analysis on the model's output by manually observing the captions generated by these 4 models.

	Vanilla CNN-LSTM	With GloVe Embeddings	With Bahdanau Attention	With Bahdanau Attention and GloVe Embeddings
BLEU-2	0.16	0.13	0.19	0.16
BLEU-3	0.05	0.037	0.11	0.06
BLEU-4	0.014	0.01	0.07	0.02
Rouge-Precision	47.10	44.70	47.22	47.07
Rouge - Recall	47.50	43.42	47.02	47.21
Rouge - F1 Score	44.62	41.43	44.46	44.52

Table 1

D. Error Analysis

In this section we selected a random sample of images from the dataset and compared the captions generated across all the models to analyze if the error metrics are a good representation of the actual model performance. In this report, we share our analysis of two random images-

Analysis 1 -

The base model in image 1 captures the presence of a man and some context like sitting on a rock overlooking the water but it doesn't accurately infer from the image. The model with glove embedding has a semantic understanding of words that captures the context of climbing but does not result in significant improvement in the caption.

The attention-based model focuses on different parts of the image and captures accurate details. It has improved the model's ability to infer in a more detailed manner. The model with Attention and Glove combined captures the general meaning of words but is phrased in a different manner which gives a completely different contextual understanding.

In conclusion, adding attention and GloVe embeddings has improved the model's performance. Also, attention has the most significant impact on the generated captions accurately.

Analysis 2 -

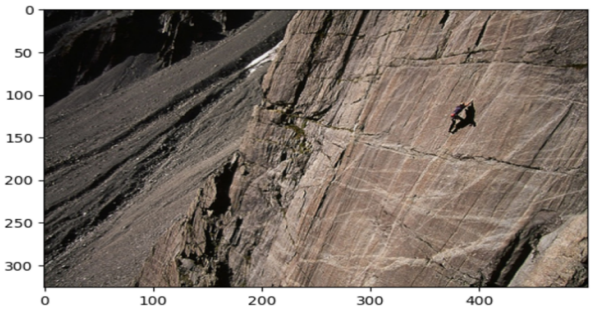
In image 2, we can infer that the base model captures the presence of a dog and a few actions but doesn't accurately describe the image. The model with pre-trained glove embedding has a semantic understanding of words but no significant results are generated compared to the base model.

The attention-based model focuses on different parts of the image capturing the presence of two dogs and the context that is playing. The attention mechanism has improved the model's ability to generate better captions.

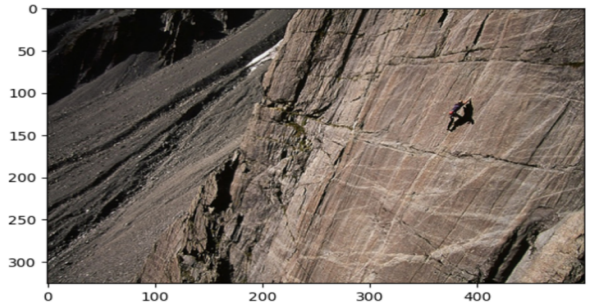
When attention and glove both combined capture the same meaning but phrases in a different manner. In conclusion, adding attention and Glove embedding improves the model performance. The attention has the most accurate caption.

Analysis 1

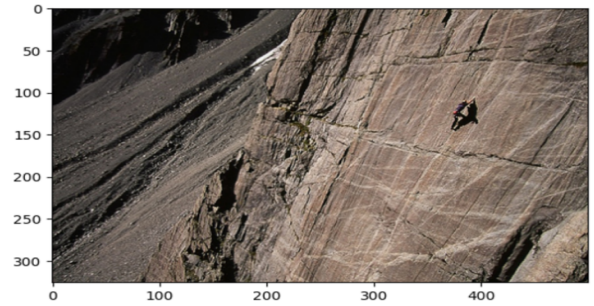
Base Model: start man is sitting on rock overlooking the water end



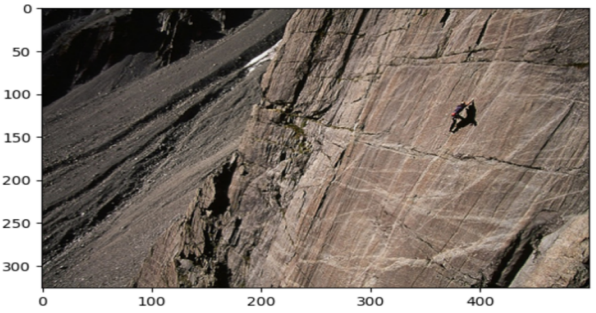
Model with Pretrain Glove Embedding:
start man is climbing rock face end



Model with Attention:start man in red shirt is climbing on rock end

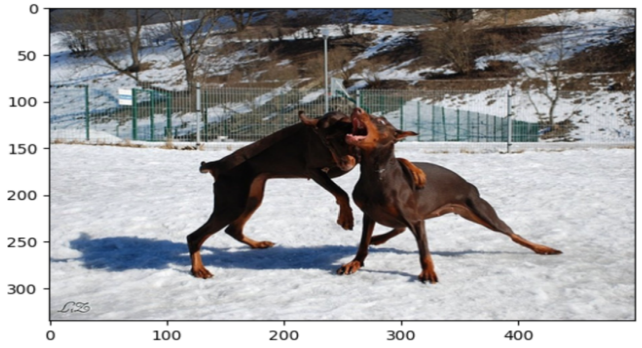


Model (Attention+Glove):start man in red shirt climbs rock face end

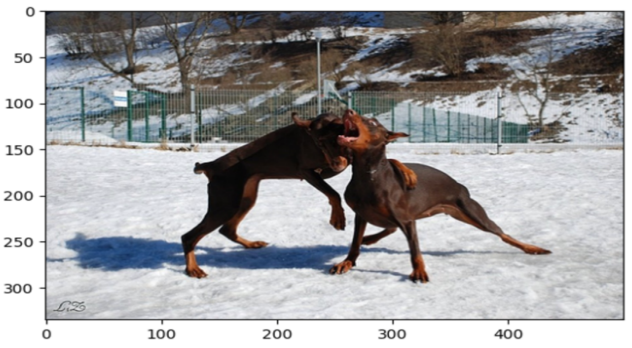


Analysis 2

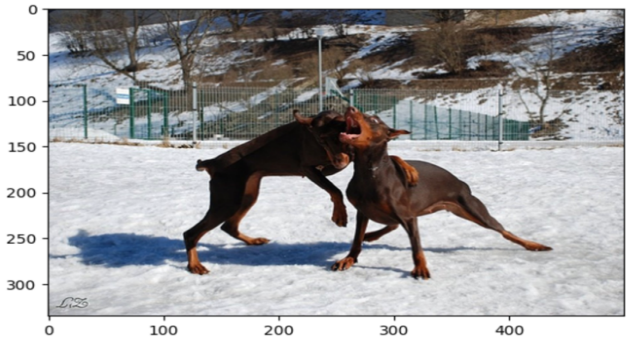
Base Model: start dog is running through the grass end



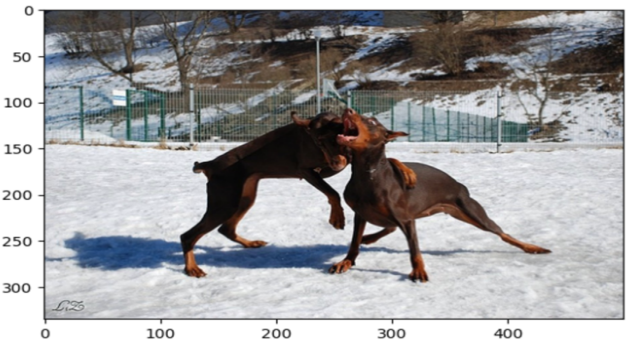
Model with Pretrain Glove Embedding:
start dog is running through the grass end



Model with Attention:start two dogs are playing in the snow end



Model (Attention+Glove):start two dogs play in the snow end



V. Conclusion

In conclusion, the image captioning model with both Attention and Glove embeddings has significant improvement compared to the base model and the model with only Glove embeddings in generating captions for images. The Attention mechanism allows the model to selectively focus on different parts of the image, capturing more context and relevant details, which helps generate more accurate and descriptive captions while Glove embeddings provide a better semantic understanding of words which helps generate captions that are more coherent and meaningful.

VI. Future Scope

The generated captions are moderately accurate as indicated by the metrics with room for more improvement. These steps can further be taken to increase the error metrics as well as to improve caption quality.

- Incorporating more advanced attention mechanisms, such as multi-head attention or self-attention could help the model capture more complex relationships between objects and actions in the images
- Fine-tuning with a larger dataset (eg. Flickr 30k) would possibly enhance the model's performance and make it more capable of generating accurate captions which is consistent with the image (eg. increasing LSTM layers).
- Currently, the generated captions are only descriptive, we could bring sentiment into the mix and can train the model on social media captions to see if the model can generate human-like captions.

VII. Contributions

Parth Maheshwari - Wrote preprocessing functions, designed the model architecture, trained Vanilla LSTM-CNN, Integrated GloVe embeddings, and wrote evaluation functions.

Vishal Kumar Panda - Designed the model architecture, incorporated the Bahdanau attention mechanism and combined both Attention and Glove embeddings into the model. Conducted error analysis on the model's predictions to evaluate and improve its performance.

VIII. References

Katiyar S, Borgohain SK. Image captioning using deep stacked LSTMs, contextual word embeddings and data augmentation. https://arxiv.org/abs/2102.11237. Accessed 22 Feb 2021

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In CVPR.

J. L. Ba, J. R. Kiros, and G. E. Hinton. 2016. Layer Normalization. (2016).

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In ACL workshop.

Fuhai Chen, Rongrong Ji, Xiaoshuai Sun, Yongjian Wu, and Jinsong Su. 2018. Groupcap: Group-based image captioning with structured relevance and diversity constraints. In CVPR. 1345–1353.

Long Chen, Zhihong Jiang, Jun Xiao, and Wei Liu. 2021. Human-like controllable image captioning with verb-specific semantic roles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 16846–16856.

Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and ShihFu Chang. 2019. Counterfactual critic multi-agent training for scene graph generation. In ICCV.

Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and TatSeng Chua. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In CVPR.

A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. CoRR, abs/1412.2306, 2014.