Image Captioning using Deep Learning Networks

Sree Krishna Suresh
Northeastern University
Boston, MA
suresh.sr@northeastern.edu

Pareekshit Reddy Gaddam
Northeastern University
Boston, MA
gaddam.pa@northeastern.edu

1 Introduction

Project-Link. Large number of images are generated each day through a wide variety of mediums such as internet, news articles and advertisements. Most of these images do not have a caption or description yet we humans can recognize them with minimum effort. But, this is not the same for a machine. Image captioning is used extensively for Content-Based Image Retrieval (CBIR) and applied in many fields such as education, military, bio-medicine and web-scraping. Also, it is used in social media platforms for identifying where a particular picture is taken, what kind of a picture it is and describe any activity that is shown in the picture which can be further used for recommending content to the user.

Image Captioning is the process of generating description of an image from the objects and actions in the image. Image captioning is a multi-modal problem that has drawn extensive attention in both the natural language processing and computer vision community.

Understanding an image mainly depends on obtaining image features. The techniques used for this purpose can be broadly divided into two categories: (1) Traditional machine learning based techniques and (2) Deep machine learning based techniques. Traditional machine based techniques involves a combination of hand crafted features like Local Binary Patterns and Histogram of Oriented Gradients. Since hand crafted features are task specific, extracting features from a large and diverse set of data is not feasible. Moreover, real world data such as images and video are complex and have different semantic interpretations. But in deep machine learning based techniques learn the features automatically and can handle a wide variety of data-sets like images and videos. Consider, Convolutional Neural Networks (CNN) that are widely used for feature learning, and a classifier such as Softmax is used for classification.

In this project, we will implement baseline model using CNN as the encoder and LSTM as a decoder and evaluate it using BLEU[6] score as a metric.

As an intermediate model, we will implement a CNN + Transformer based model using Attention Mechanism for Caption Generation and evaluate it using BLEU[6] score as a metric.

For the final model, we investigate the effect of pre-trained embeddings on the task of image captioning by BERT context vectors [12] to enhance the models performance and reduce training time.

The task is divided into two modules, namely Image base and Language based module. Image based module extracts the features and nuances out of the image and the Language based module translates the features and objects from the image to a natural sentence. For the image based module to encode the image we used CNN and for the decoder we used Pre-Trained BERT along with LSTM.

2 Literature Survey

Despite challenges, the problem of image captioning has achieved significant improvements over the past few years. There are several research works pertaining to image captioning. Research paper [1] focuses on using CNN-RNN based, CNN-CNN based and Reinforcement-based framework to caption

images and summarize the benefits and challenges. Briefly talks about the benefits such as Intelligent monitoring, Human-computer interaction, Image and Video annotation and major challenges like Richness of image semantics, Inconsistent objects during training and testing and Cross-language text description of images.

Research paper[2] does a A thorough review of models, evaluation metrics, and data-sets on image captioning. This paper explains the different image caption methods like the search-based approach, the language template-based approach, and the sequence-based approach using neural networks, and focuses on the refinement of the sequence-based approach. The paper mentions several evaluation measures and summarized their advantages and disadvantages. [3] Developed a unified model which can be fine tuned for either vision-language generation (e.g., image captioning) or understanding (e.g., visual question answering) utilizing specific self-attention masks for the shared transformer network tasks. They generated a model that is pre-trained on large amounts of image-text pairs based on two objectives: bidirectional and seq2seq vision-language prediction.

Research paper[4] presents a model architecture based on Inception-ResNetv2 for image-feature extraction and Transformer for sequence modeling which achieves the best performance when trained on the Conceptual Captions dataset. Presented a new dataset called Conceptual Captions. It has many important characteristics with around 3.3M examples which is larger than the COCO image-captioning dataset. The dataset includes a wide variety of images, including natural images, product images, professional photos, cartoons, drawings. The captions in the dataset are based from original Alt-text attributes, automatically transformed to achieve a balance between cleanliness, informativeness, and learn-ability.

Reseach paper[7] Provides an In-depth overview of Deep Learning for Image Captioning. It briefly talks about deep learning based methods such as Supervised learning, reinforcement learning, and GAN based methods that are commonly used in generating image captions. The categories of supervised learning based methods like Encoder-Decoder architecture-based, Compositional architecture-based, Attention-based, Semantic concept-based, Stylized captions, Dense image captioning, and Novel object-based image captioning are also discussed.[8] This paper mentions and highlights that the visual and linguistic modalities for caption generation need not be jointly encoded by the RNN. It also presented a systematic evaluation of a number of variations on architectures for image caption generation and retrieval. The main focus was on the differences between terms ‘inject’ and ‘merge’ architectures.

3 Method / Study

3.1 Baseline Model: CNN-LSTM Architecture

For our baseline model, we used flickr8k dataset and a inject model which combines the encoded form of the image with each word from the text description generated so-far. This approach uses the recurrent neural network as a text generation model that uses a sequence of both image and word information as input, in order to generate the next word in the sequence. The architecture has two inputs, namely the image vectors and the word embeddings of the captions. The embedding vector is passed through an LSTM architecture which will learn how to make the appropriate word predictions. The image vector and the LSTM predictions are combined together as one unit and passed through few dense layers and a SoftMax activation function, which is equivalent to the size of the pre-defined vocabulary.

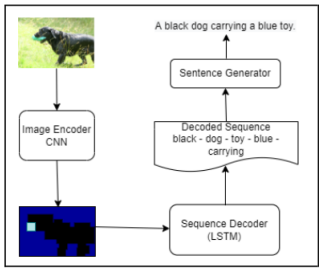

Figure 1: Encoder-Decoder / Inject Architecture

3.2 Intermediate Model: CNN-Transformer Architecture

As an improvement over the baseline model, we tried implementing the transformer model[12] by dispensing recurrence and instead relying solely on attention mechanism to efficiently caption the images with parallelization.

The Transformer is a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer architecture allows for significantly more parallelization and can reach new state of the art results in translation quality.

The transformer model employs an encoder-decoder architecture similar to that of an RNN. The main difference is that transformers can receive the input sentence/sequence in parallel, i.e, there is no time step associated with the input, and all the words in the sentence can be passed simultaneously.

For Image Captioning, we first did image feature extraction model using InceptionV3. We only need to extract an image vector for our images. Hence we removed the softmax layer from the model. We mapped each image name to the function to load the image and pre-processed each image with InceptionV3 and cache the output to disk and image features are reshaped to 64×2048. We later extracted the features and stored them in the respective .npy files and then pass those features through the encoder.

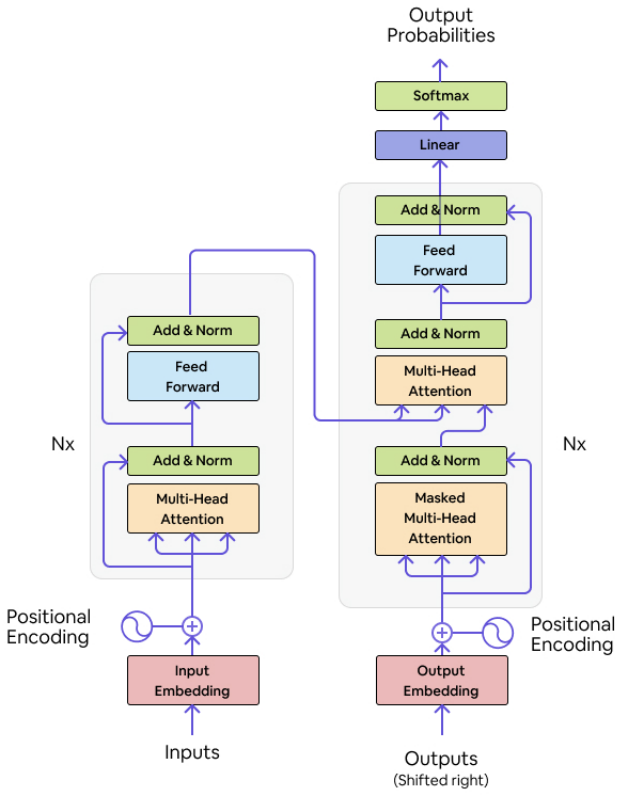

Figure 2: The Transformer - model architecture[12]

We then tokenized the captions and built a vocabulary of all the unique words in the data. We will also limit the vocabulary size to the top 5000 words to save memory. We will replace the words that are not in vocabulary with the token < unk >

The transformer model has two modules. The encoder module and the decoder module.The encoder block has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a

simple, position-wise fully connected feed-forward network. Multi-headed attention in the encoder applies a specific attention mechanism called self-attention. Self-attention allows the models to associate each vector in the input, to other vectors. Multi-head attention allows the model to jointly attend to information from different representation sub spaces at different positions. With a single attention head, averaging inhibits this[12].

In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. The attention vectors of the captions and the attention vectors of images from the encoder are passed into second multi head attention.

This attention block will determine how related each caption vector is with respect to each other. This is where the image to caption mapping takes place. The decoder is capped off with a linear layer that acts as a classifier, and a softmax to get the word probabilities.

In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between.

3.3 Final Model: CNN-LSTM Architecture + BERT Embeddings

For the final model, we planned to test the effect of pre-trained embeddings for image captioning by BERT context vectors to enhance the models performance and reduce training time.

We used the pretrained CNN model (ResNet-101 CNN) as our encoder to focus on enhancing the performance of the decoder. We removed the max pooling and the last two linear layers as we require the image features, after which we injected the output of the modified ResNet into an adaptive pooling layer to create a fixed size output vector to later on to be passed into the decoder.

Bidirectional Encoder Representations from Transformers uses a Transformer to generate a bi-directional contexualized word embeddings conditioned on the context of the word in a sentence. BERT uses two different modules. Namely, BERT base and BERT large. We tried implementing the base version of the BERT model that has 12 encoder layers in the Transformer, 768 hidden units in the fully connected neural network and 12 attention heads. We did not implement the large model due to the increase in training time with that model.

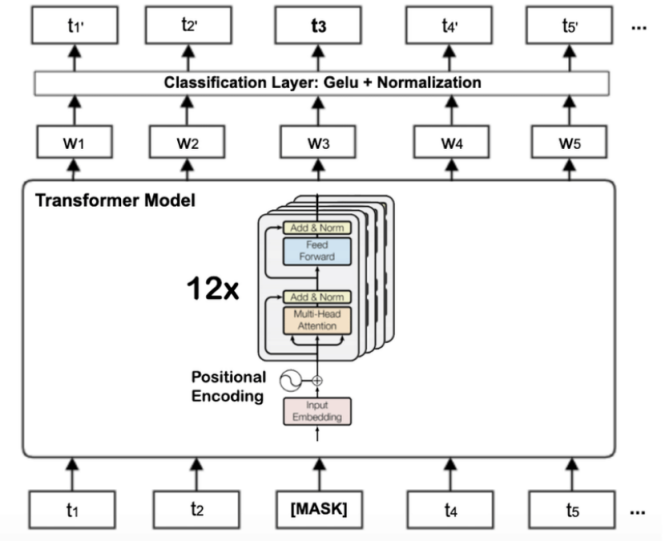

Figure 3: The Transformer based BERT base architecture with twelve encoder blocks [13]

In our model, a long short-term memory (LSTM) network is used. It generates the caption, one word at a time, by taking inputs as the context vector of the current step, the previous hidden state, and the previously generated words represented as Bert embeddings. Bert embeddings help in choosing the next word by considering the context of the previous words.

4 Experiments

4.1 Benchmark Datasets

Dataset	Images	Vocabulary size	Max Length
Flickr30k	31783	22188	80
Flickr8k	8091	8357	92

Table 1: Data-set Vocabulary Description

We plan to use two open-source datasets for the project:

- Flick8k Dataset[10] - 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events. The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations.
- Flickr30k Dataset[5] - 158k captions with 244k coreference chains, linking mentions of the same entities across different captions for the same image, and associating them with 276k manually annotated bounding boxes.

4.2 Evaluation Metrics

The evaluation metrics used for our baseline models are accuracy and loss while training the model, 'categorical cross-entropy ' in the last layer to predict the probability of each word.

In order to evaluate the captions we used The Bilingual Evaluation Understudy (BLEU) method [6] is adopted to evaluate the quality of translated sentences in machine translation. It compares each translation segment with a set of reference translations with good translation quality and calculates each segment score then estimates the overall quality of the translation. In the field of image description, as a similarity measurement method, BLEU adopts an ***n-gram*** matching rule. The BLEU evaluation metric can be evaluated by analyzing the co-occurrence frequency of ***n-gram*** in the predicted caption and the label. Let Candidates and Reference be the predicted caption and the label respectively. For an n-gram, the precision of the sentence can be expressed as follows:

$$P_n = \frac{\sum_{C \in \{candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{candidates\}} \sum_{n-gram' \in C'} Count_{clip}(n-gram')} \tag{1}$$

where $Count_{clip}(n-gram)$ represents minimum number of occurrences of $n-gram$ in the Candidates and the reference. $Count(n-gram')$ indicates number of times $n-gram$ appears in Candidates.
As the number of tuple increases, the P_n of the $n-gram$ statistics will decrease. To prevent the training results from tending to short sentences, multiply by the brevity penalty factor to get the final evaluation formula:

$$BLEU = BP * exp(\sum_{n=1}^{N} w_n \log P_n) \tag{2}$$

where w_n is the weight of the precision of $n-gram$. Let c be the length of the Candidates and r be the reference corpus length. The value of BP is as follows:

$$BP = \begin{cases} 1, & \text{if } c < r \\ e^{1-r/c}, & \text{if } c > r \end{cases} \tag{3}$$

Specifically, BLEU-1 divides the description sentence and the label into words, counts the number of times the words in the description sentence appear in the label one by one, and records the minimum

number of times that tuple appears in the description sentence and label, and Carry out the ratio with the description sentence, and finally, to avoid the bias problem of the generated description sentence being too short, multiply it by a penalty factor to get the final result. BLEU-2 divides the description sentence and label into 2-tuples containing two words for statistics and calculations. For our model we are calculating up to 4-tuples.

4.3 Baseline Results

For our baseline model, we used flickr8k dataset. We used a inject model which combines the encoded form of the image with each word from the text description generated so-far.This approach uses the recurrent neural network as a text generation model that uses a sequence of both image and word information as input in order to generate the next word in the sequence.The architecture is defined as follows, for this model, we made use use of two inputs, namely the image vectors and the word embeddings of the captions, for making the predictions. The embedding vector is passed through an LSTM architecture which will learn how to make the appropriate word predictions. The image vector and the LSTM predictions are combined together as one unit and passed through some dense layers and a SoftMax activation function, which is equivalent to the size of the pre-defined vocabulary.

For dealing with each of the images, we made use of inception V3 transfer learning model to convert the images into an appropriate vector representation. The final SoftMax function layer of inception V3 was excluded and we used 2048 features for performing accurate vector encoding. For the image captions We updated all the letters to a lower case format and removed punctuation's from them. The captions were stored in a dictionary for respective images. To render the word embedding vectors for each unique word for a fixed length, every single index of words is mapped to a 200 length vector to make it an uniformly sized distribution. Pre-trained glove.6B.200d.txt[11] was used for this model to create an embedding matrix.

Hyperparameter	**Value**
Epochs	50
Batch size	64
Learning Rate	$1e^{-4}$
Optimizer	Adam

Table 2: Base line Experimental Setup

Using flickr8k[8] dataset, we trained the above architecture Table 4.3 on images of size $299 * 299$, with inception V3 layers in the beginning to learn the features of the images, followed by dense, dropout, embedding, dense, LSTM layer and dense layer. This architecture resulted in 75% accuracy after 50 epochs (Figure 1) and we obtained the as shown in Table 4.5.

4.4 Transformer Model Setup

For the transformer model we are using Sparse Categorical Crossentropy loss. Adam optimizer for optimizing the model and updating the weights of the network Adam Optimizer achieves as it has greater efficiency and faster convergence. We are using a custom learning rate scheduler to vary the learning rate with every 4000 steps. Below is the hyperparameter setupTable 4.4.

Hyperparameter	**Value**
Epochs	30
Batch size	64
Learning Rate	varying
Optimizer	Adam

Table 3: Transformer Experimental Setup

4.5 Final Model: CNN-BERT Model

Considering the fact that the BERT model will outperform the previous architectures we have used by the fact that BERT uses contexualized word embbedings but we were surprised by the margin by which it outperformed the previous models. By this we can understand that the use of a word in a sentence with the proper context is really important in generating captions. It makes much sense because captions are supposed to relate to objects in the image.The model was optimized using the below hyper-parameters obtained

Hyperparameter	Value
Epochs	4
Batch size	32
Learning Rate	0.0004
Optimizer	Adam
Attention Dimension	512
Embedding Dimension	768

Table 4: Bert Experimental Setup

Figure 4: Accurate captions by the BERT model

The predictions made by the BERT model are very interesting as the captions generated use different words in the same context to imply the same meaning. Which implies that the model is actually learning the caption generation process rather than simply replicating it. In the Above figure, the word "back window" was changed to "back" and the the word "holidays" is changed to "Christmas" which is interesting as the model is meeting the goal.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
CNN-RNN	0.452381	0.481360	0.507511	0.439849
CNN-Transformer	0.389400	0.550695	0.632582	0.654890
CNN-BERT	0.402150	0.526101	0.781502	0.654890

Table 5: BLEU Scores

5 Conclusion

We have implemented the base model to get baseline scores on a small dataset. We also implemented our model which uses multi-head attention transformer instead of soft-attention. The result was better than the baseline but improvements can be made to get better results. So BERT based model was developed and it performed better than the baseline and transormer model.Also we have used Inception V3 and Resnet101 and the results can be compared to understand which CNN architecture is best suited for our model. Our experiments outline the importance of word embedding in nature language processing and offer a new method in integrating BERT with already developed models to enhance their performance. Possible extensions to our work would be to train a new model with BERT large as apposed to BERT base, utilize beam search in validation, and train the models until the training loss converges.

References

[1] Liu, Shuang & Bai, Liang & Hu, Yanli & Wang, Haoran. (2018). *Image Captioning Based on Deep Neural Networks*. MATEC Web of Conferences. 232. 01052. 10.1051/matecconf/201823201052.

[2] Luo, Cheng, L., Jing, C., Zhao, C., & Song, G. (2022). *A thorough review of models, evaluation metrics, and datasets on image captioning.* IET Image Processing /, 16(2), 311–332. https://doi.org/10.1049/ipr2.12367

[3] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso & Jianfeng Gao (2019) *Unified Vision-Language Pre-Training for Image Captioning and VQA.* arXiv.1909.11059

[4] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. (ACL 2018). *Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning.* https://aclanthology.org/P18-1238

[5] Hsankesara, A. (2018). Flickr Image dataset [https://www.kaggle.com/hsankesara/flickr-image-dataset].

[6] Papineni, K., Roukos, S., Ward, T., Zhu, W. *Bleu: a method for automatic evaluation of machine translation.* In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics

[7] Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, Hamid Laga (2018) *A Comprehensive Survey of Deep Learning for Image Captioning*. arXiv:1810.04020

[8] Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, Barbara Plank (2016) *Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures*. arXiv:1601.03896

[9] Marc Tanti, Albert Gatt, Kenneth P. Camilleri (2018) *Where to put the Image in an Image Caption Generator*. arXiv:1703.09137

[10] Adityajn, A. (2020).Flickr 8k Dataset [https://www.kaggle.com/datasets/adityajn105/flickr8k].

[11] Jeffrey Pennington, Richard Socher, and Christopher Manning. (2014) *GloVe: Global Vectors for Word Representation*(https://aclanthology.org/D14-1162) EMNLP,Doha, Qatar.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017) *Attention Is All You Need. arXiv preprint arXiv:1706.03762*.

[13] Khalid, Usama & Beg, Mirza & Arshad, Muhammad. (2021).*RUBERT: A Bilingual Roman Urdu BERT Using Cross Lingual Transfer Learning.* arXiv:1810.04805