A Deep Learning Model to Generate Image Captions

Shashank Parmar, Raman Tyagi, Prince Kumar Dhankar

Department of Software Engineering, Delhi Technological University, New Delhi, India

DOI: https://doi.org/10.51244/IJRSI.2023.1011054

Received: 21 November 2023; Revised: 28 November 2023; Accepted: 04 December 2023; Published: 23 December 2023

ABSTRACT

How computers can automatically describe the substance of photographs using human language is a topic that interests us greatly. We choose to use the most advanced image caption generator currently available to obtain a deeper understanding of this computer vision topic. Show, attend and tell Visually attentive neural image caption generator [12]. Our machine learning-based neural network picture description generator is created in Python using the Pytorch ML framework. In our workflow, we've determined five key elements: Data preparation, a **convolutional neural network (CNN)** it helps in encoding, a **recurrent neural network (RNN)** it helps in decoding, a beam search to determine the best description, generation of sentences, and assessment make up the **R1-R6 framework**. The quality and correctness of the generated caption are evaluated using the BLEU-4 score. Each member of our group contributed equally to moving the project forward as we distributed the five elements mentioned above equally among ourselves. All five components have been completed successfully, and we can now use Kaggle Notebook to train our network. After the network has been trained and is performing satisfactorily, we continue to see the attention mechanism.

INTRODUCTION

The contemporary landscape of computer vision and machine learning is increasingly focused on endowing machines with the ability to autonomously generate textual descriptions for images. This multifaceted endeavor encompasses the recognition of intricate photographic scenes, involving the discernment of salient features and the transformation of visual depictions into coherent and simplified language. The potential of this undertaking manifests notably in the development of assistive technologies for individuals with visual impairments and the streamlining of automated captioning tasks in online environments.

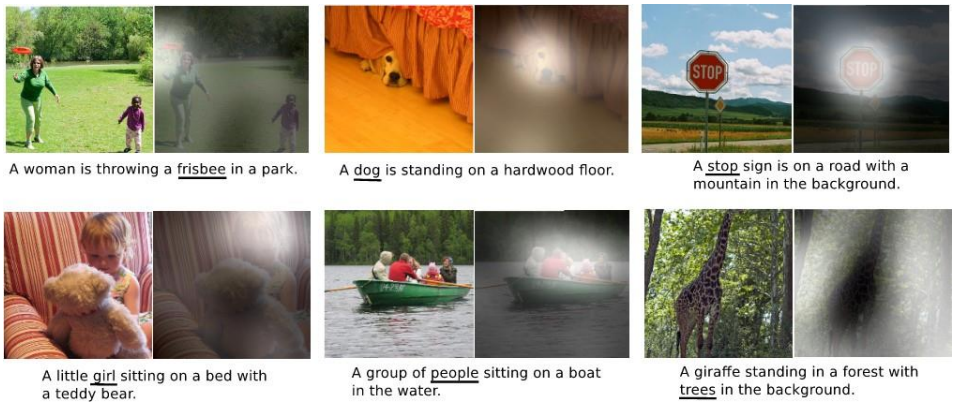

Figure 1. This figure is from Show, attend and tell visualization (adapted from [12])

While the pursuit of automatic image description has been a subject of considerable attention in recent decades, extant research publications have grappled with various challenges, including grammatical inaccuracies, cognitive incongruities, and the inclusion of extraneous content [5]. However, propelled by remarkable advancements in neural networks, recent endeavors have turned towards the integration of convolutional and recurrent neural networks to address these issues, yielding promising outcomes [2].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

Noteworthy contributions in this domain advocate the utilization of a deep convolutional neural network (CNN) in tandem with a recurrent neural network (RNN) to fulfill the objectives of image description. Building upon this foundation, a subsequent publication enhances the methodology by introducing an attention mechanism. Illustrated in Figure 1, the incorporation of a learnable attention layer empowers the network to selectively focus on specific regions of an image corresponding to each generated sentence. This augmentation represents a pivotal advancement, elevating the precision and contextual relevance of the generated textual descriptions. This discourse aims to synthesize and contribute to the ongoing dialogue within the scholarly community, offering insights into the evolution of methodologies for automated image description in the era of sophisticated neural network architectures.

METHODOLOGY AND ARCHITECTURE

To convert unformatted raw input data (including images and captions), we have created data preparation scripts; using a higher dimensional vector space to extract and encode picture characteristics using a pre-trained CNN architecture; a decoder that uses an LSTM-based recurrent neural network to translate encoded information into descriptions in natural language; Beam Search is used to determine the caption with the highest likelihood. For improving overall accomplishment, RNN uses an attention mechanism that allows it to recognise features from a part of the input image that has been deliberately emphasized. Below, a detailed explanation of each element of our generator pipeline will be provided.

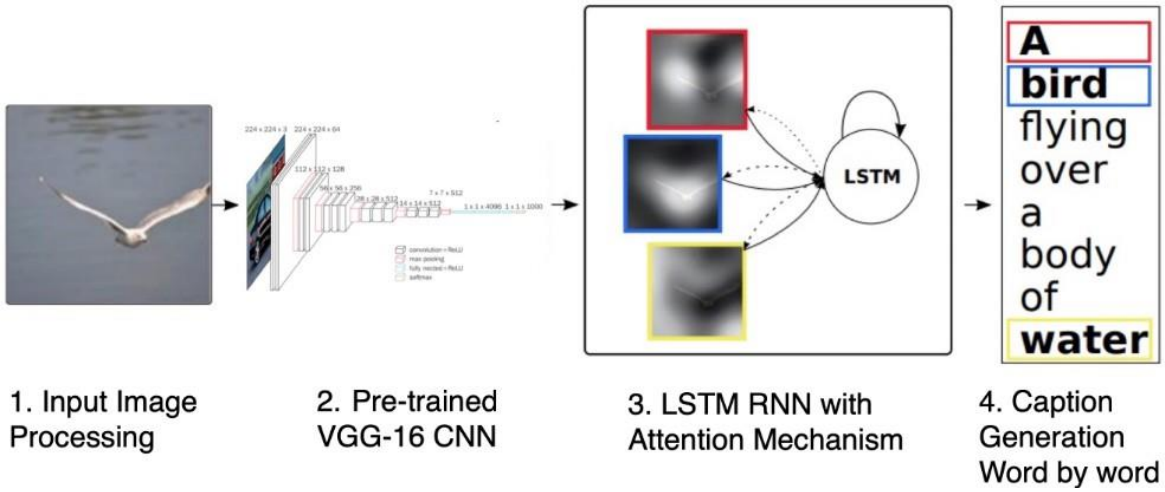

Fig 2. Picture depicts the photo's description generation process.

Data Sources

The COCO dataset [6], Flickr30k [8], and Flickr8k [3] are the three most famous picture caption training datasets in the Computer Vision research field. These datasets, which each contain 123,000, 31,000 and 8091 captioned photos, each have 5 descriptions for each image. Due to our limited storage and processing capability, we currently choose the Flickr8k dataset, which has the fewest photographs, as our main data source over the other two.

Data Preprocessing

Captions and images form the input data, thus should be preprocessed to place them in the correct formats for encoder and decoder, respectively. We must convert photos into the appropriate format because our image caption generating pipeline relies on a modern CNN network that has been pre-trained

PyTorch documentation states that pre-trained vgg-16 models assume input images to be normalized to fall between [0, 1] and take the form of 3-channel RGB photos with the shape (3, H, W), where H and W are

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

assumed to be at least 224. Accordingly, data preprocessing comprises importing and scaling image data into (N, 3, 256, 256) dimensions as well as normalizing pixel values to be between [0, 1] with a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225].

Convolutional Neural Network

In order to later feed these visual qualities to RNNs, Prior to encoding, the CNN should first draw out graphical properties of various dimensions. VGG-16 & Res Net are frequently suggested image encoding algorithms. The PyTorch toolkit's pre-trained vgg-16 model is selected.

CNN is used in this project to encode the features rather than classifying the photographs. As a result, the network's end deleted the max pool and entirely connected layers. The dimensions of the supplied photo matrix are (N, 3, 256, 256), while the outcome of this new structure has (N, 14, 14, 512) dimensions. We also attached an adaptive 2D layer to the encoder's architecture to accommodate input photos of various sizes.

To cut down on computing costs, we removed gradient in our picture captioning architecture. We might get a better overall performance with some fine-tuning.

Soft Attention Mechanism

The next word in the description is produced by the network by following instructions from the attention mechanism to concentrate on a particular region of the image. The CNN outcome and the older state, which is upgraded after each reiteration, were combined to choose the attention region.

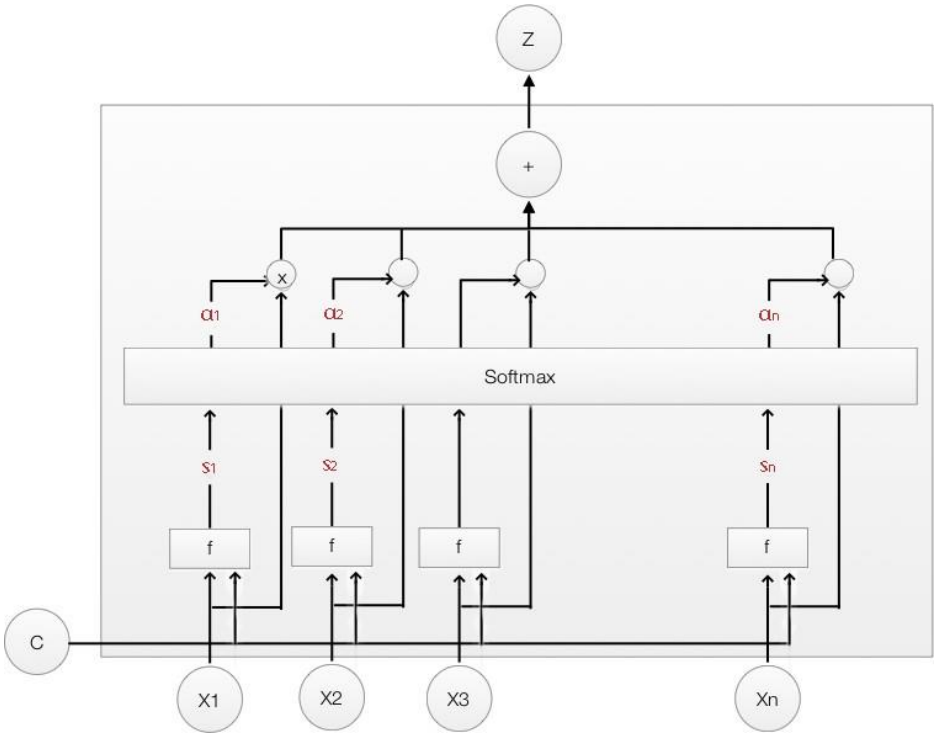

Figure 3. A demonstration of soft attention model [1]

When creating the "attention layer", we linearly transformed the encoder output as well as the historical state output. Total of the linearly activated outputs is the activation function, which is ReLU. The attention area (applying alpha to encoder output) and weight (the value of alpha) are returned.

The attention mechanism can be trained by back-propagation, in contrast to CNN without fine-tuning.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

Later, a decoder using RNN uses the returned attention area.

Recurrent Neural Network (Decoder)

Using an RNN-LSTM that can produce words in a sequential manner, the decoder must construct word-by-word photo descriptions. The CNN photo features vectors that were formed during the data preprocessing stage and the encoded picture captions serve as the input for the decoder.

The RNN consists of an attention module we drafted and built, an LSTM cell module, & an LSTM cell module, in addition to four completely linked layers given by the PyTorch library for initializing the states of the LSTM cell and term dictionary.

As we get encoded photos and captions, we initially sort them according to the image's encoded key length. We only want to analyze the encoded photographs whose caption lengths are at least the same as the number of iterations in order to improve productivity and decrease time of training.

In order to create the attention-masked photos with a particular area highlighted, we first feed the encoded photos and the historical state of the LSTM network into the "Attention module" in each iteration of the LSTM network. Next state of the LSTM is obtained by concatenating the attention-grabbing visuals with all of the previous words' embedded captions. Fully linked layers can then forecast the likelihood of the present word embedding based on the present state & add to the word embedding prediction matrix.

Beam Search

The Beam Search in the last step of the Show and Tell paper, identifies the statement that has the greatest chance of recurrence given the input photo. The algorithm repeatedly considers the set of the top k sentences up to time a as candidates to generate sentences of size a + 1, being an informed search method, and keeps only the top k of them because this more closely resembles the chance of obtaining the highest cumulative probability score as cited in the paper. We applied beam search sizes with values 1, 2, 3, 4, 5, & the evaluation of the *Bilingual Evaluation Understudy* score appears to be more favorable for beam sizes 3 and 4. Therefore, we use 3 as size for beam search.

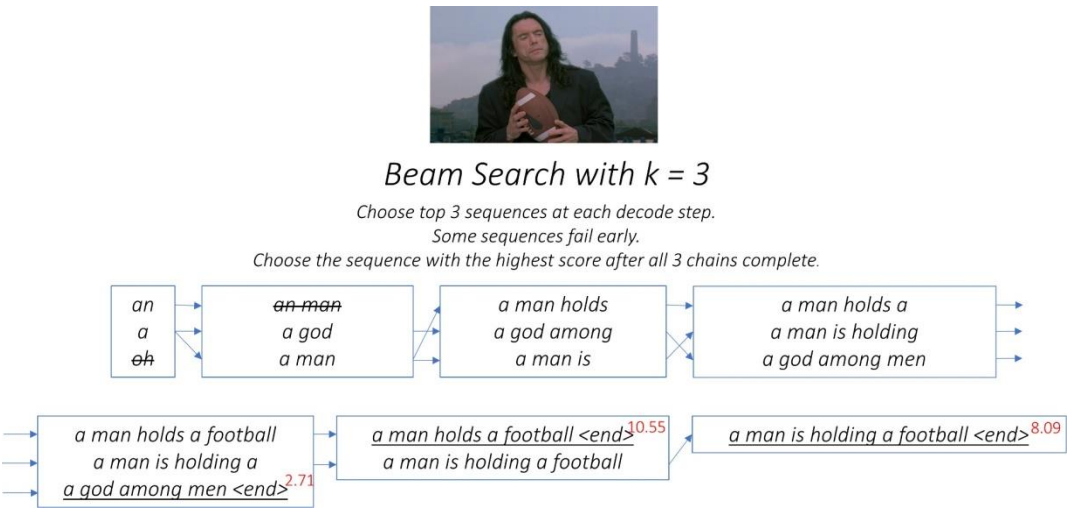

Fig 4. An example of Beam Search in our sentence generation.

From a github project, Figure 4 provides a useful illustration graphic [10]. This diagram exemplifies how beam search chooses the set of words with the global maximum rather than word with the highest probability at each stage (local maximum).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

Loss Function

As a series of probabilities of word occurrences make up the nature of our RNN output, we advise utilizing cross *entropy loss* to assess the *output's quality*. When a "classification model" generates a probability value between *0 and 1*, the most common and practical method to assess how well it is doing is to look at this probability value.

$$E_{entropy} = -\sum_{n}^{N} t_k^n \ln p_k^n \qquad (1)$$

EXPERIMENTS

In the course of our experimental investigations, numerous hyperparameters underwent systematic manipulation to refine the performance of our model. Initially, a learning rate of 4 × 10^-4 was applied during the training phase, resulting in significant oscillations in the training loss. To mitigate this issue, we implemented an adaptive learning rate schedule, denoted as the "lr scheduler," which incrementally elevated the learning rate by a factor of 0.99 after each epoch. This strategic employment of the scheduler effectively alleviated the pronounced oscillations, contributing to enhanced training stability.

Furthermore, the adjustment of the batch size constituted a pivotal aspect of our experimentation. Commencing with a batch size of 5 proved suboptimal, given the extensive composition of the dataset—specifically, the presence of hundreds of images within the "flicker 8k" dataset. This diminutive batch size hindered convergence of the loss function. Consequently, a progressive increase to a batch size of 50 was implemented, yet this expansion surpassed the 12-gigabyte (GB) memory constraint imposed by Kaggle's graphics processing unit (GPU). Ultimately, a judicious compromise was reached, and a batch size of 32 was selected, remaining within the GPU memory limitations while concurrently ensuring sufficient convergence of the loss function. This iterative optimization process underscored the necessity for a balanced configuration of hyperparameters to achieve both computational efficiency and optimal model performance.

EVALUATION

Although the pipeline has been established, which is not apparent whether it is entirely accurate or whether it can be taught in the manner for which it was designed. Our model must then be properly trained in order for it to learn. Another significant portion of this project that is yet unfinished is (R6) if the model has proven to be accurate. Sentence generation and performance assessment are the greatest ways to show performance of our description generator.

Evaluation Metrics

The *BLEU-1, BLEU-2, BLEU-3, and BLEU-4* metrics are the most often used to assess the level of accuracy of text creation in natural language processing tasks. Here, our main evaluation parameter was the 4-gram *BLEU* score *(BLEU-4)*. The following stage is to look into additional metrics (such METEOR) described in pertinent studies [11, 12] and attempt to make improvements to our model to make it more accurate.

Model Performance

Today, we have a 9.8 BLEU-4 score within 35 epochs (BLEU-4 is within range 0 to 100). Our model performs less well than state-of-the-art caption generators, as seen by the 19.5 BLEU-4 score reported in *Show, Attend, and Tell* on the Flickr 8k. We made the decision not to spend too much time and effort simply increasing our BLEU-4 score to match the findings in article,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

RESULTS

Data is translated into numbers using a lexicon of words, as was mentioned before. When data has ingested, the pipeline will also produce output in the encrypted format that must first be converted into English for human comprehension. The outcome of RNN is a series of word likelihoods, which is another crucial aspect. Selecting the word with the most likelihood at each decode stage in recurrent neural networks frequently yields less-than-ideal results. Beam Search, a popular method for determining the best pathways for decoding words in natural language, is used. which was covered in Section 2.6. The subtitles for a few test images can be found below.

Fig 5. An illustration of a caption that was created incorrectly.

Evidently, several of the automatically produced captions on the network omit significant image details, while others misidentify specific visual components. For instance, “boy in blue shirt is holding red pole in front of playground” can be seen in Figure 5’s top left image. The shirt in the image is identified by the model but the model fails to identify the correct color.

We were fascinated by how regularly and skillfully descriptions for pictures of dogs catching Frisbees were generated in test photographs. This is most likely because there are so many pictures showing dogs in training jumping to catch Frisbee. Furthermore, it’s probable that the pre-trained CNN is very familiar with how a dog looks.

Generated Captions

As seen in Figures 6, 7, 8, and 9, we were successful in producing captions that largely followed proper grammar rules and were legible by humans. The majority of the images give an accurate description of the objects in the scene, count how many times they appear, and give a logically sound verb to logically finish the sentence. Also effectively recorded are the object’s colors and the spatial interactions between items. Then, CNN is able to record the characteristics of items. Finally, conjunctions will be used by RNN and Attention Mechanism to logically join these words to form a whole phrase.

Fig 6. Dog jumping through an obstacle course

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

Figure 7. Brown dog is running through the water

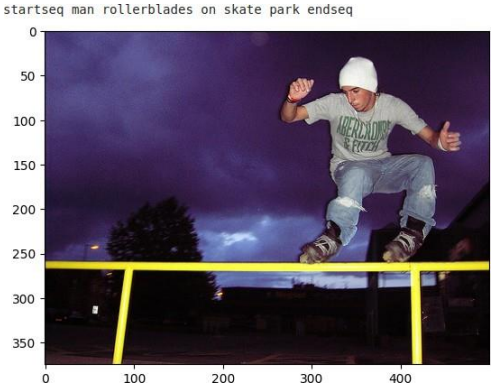

Fig 8. Man roller blades on skate park

Fig 9. Child in swimming pool

At the conclusion of this report, we have additionally added additional results (Figures 10 and 11).

Attention Mechanism Visualization

The capacity to visualize attention mechanisms enables us to comprehend where the area of the image is being focused when creating a certain phrase, which is crucial for enhancing the network's potential to grasp scenes. Each phrase in our visualization was made by layering a see-through mask over the original photo at each RNN node. As a result, the dark zone had little effect on the phrase that was formed, and the light region could be used to indicate the focal area.

CONCLUSION AND FUTURE WORK

Because mechanized picture descriptioning is still in its babyhood, there are various research projects now underway focusing on extra precise photo characteristic extraction and semantically superior phrase synthesis. We were only able to complete the tasks listed in the project proposal due to limited

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

computational resources, and we had to use a smaller dataset (Flickr 8k). There might be enhancements if additional time is given. Because we used a pre-trained CNN network straight into our pipeline without making any further adjustments, the network did not respond to this specific training dataset. In order to enable fine-tuning and experiment with other CNN pre-trained networks, we anticipate getting a somewhat higher BLEU-4 score. Training on a mix of Flickr_8k, MSCOCO and Flickr_30k could also result in improvements. In general, the output will be more accurate the more divergent the training dataset. We can all agree that this study has piqued our curiosity in the potential applications of machine learning to computer vision, and we anticipate exploring these further in the future.

ACKNOWLEDGEMENT

Since we started the project from scratch, we ran into many execution challenges, many of which had to do with *PyTorch* syntax and usage conventions. With assistance from the *PyTorch developer community* and by consulting some pertinent open *Github* sources, we were able to resolve these difficulties.

Fig 10. Our caption generator produced image and caption pairs.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

Fig 11. By using our image description generator, we can produce photos and captions both.

REFERENCES

1. Jonathan Hui Blog. https://jhui.github.io/2017/03/15/Soft-and-hard-attention.
2. Alex Graves. Generating sequences with recurrent neural networks. CoRR, abs/1308.0850, 2013.
3. Micah Hodosh, Peter Young, and Julia Framing image description as a ranking task: Data, models and evaluation metrics. J. Artif. Int. Res., 47(1):853–899, May2013.
4. Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image IEEE Trans. Pattern Anal. Mach. Intell., 39(4):664–676, Apr. 2017.
5. Polina Kuznetsova, Vicente Ordonez, Alexander Berg, Tamara L. Berg, and Yejin Choi. Collective generation of natural image descriptions. pages 359–368, 2012.
6. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll´ar, and C. Lawrence Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 740–755, Cham, 2014. Springer International Publishing
7. S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. pages 730–734, Nov 2015.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume X Issue XI November 2023

8. Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence Int. J. Comput. Vision, 123(1):74–93, May 2017.
9. PyTorch. https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html.
10. Sagar https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
11. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Show and tell: A neural image caption generator. CoRR, abs/1411.4555, 2014.
12. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S.Zemel, and Yoshua Show, attend and tell: Neural image caption generation with visual attention. CoRR, abs/1502.03044, 2015.