Automated Image Caption Generator with LSTM and CNN

Aryan Shrestha
Georgia State University
Atlanta, GA
ashrestha6@student.gsu.edu

Abstract

The main goal of this project is to create a model which can automatically generate a description of the image provided to it. This model will be based on concepts of CNN, RNN, LSTM along with attention mechanism. CNNs are deployed for extracting spatial information from images. RNNs are harnessed for generating sequential data of words. LSTM is good at remembering lengthy sequences of words. The dataset is a combination of images and their text descriptions provided by flickr8k. The implementation part will mainly deal with object detection, text prediction and tokenization. Beam search is used to achieve a better BLEU score and overall model performance. Some of the use cases are making google search easy, aid to the blind, classifying pictures, caption bot etc.

1. Introduction

Image caption has increasingly caught the attention of many researchers in the field of artificial intelligence due to the quick growth of AI in recent years and it has grown to be a challenging task. Automatically generated image captions based on language descriptions that correspond to the content shown in an image are a crucial component of scene comprehension, which combines expertise in both computer vision and natural language processing. It requires both methods from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order. Recently, deep learning methods have achieved state-of-the-art results on examples of this problem. A single end-to-end model can be defined to predict a caption, given a photo, instead of requiring sophisticated data preparation or a pipeline of specifically designed models. The use of image captions is widespread and important, for the development of human-computer interaction, for instance.

The dataset contains pairs of images and their corresponding captions. Each image has five captions. The encoder-decoder architecture is combined with attention mechanism. The encoder is used to extract image representations and the decoder is used to generate image captions. The evaluation metric used is BLEU-4. Beam search is implemented to generate better image captions.

2. Related Work

Recently, several methods have been experimented with for automatic image caption generation. In [1] one of the most intriguing and pragmatic neural model is introduced which combines CNN and LSTM model into a hybrid model. This model is specifically designed for sequence prediction problems for spatial inputs such as images. It is state of the art neural model for NLP tasks such as transformer for sequential image data. The powerful CNN model can be used for sequential natural language data. Their work has encouraged a lot of researchers to use different architectures of the CNN-LSTM model. They got some partial errors due to lack of attention to specific details in images. They demonstrated the effect of emitted words on hidden states in LSTM and found out that semantically close emitted words such as plate and bowl result in similar movements despite different previous context and divergences occur only upon emission of semantically far words such as vase and food. Their work also suggests that the attention-mechanisms may yield certain improvements to this task.

Similarly [2] suggests that the basic RNN design struggles with longer sequences, but a special variant in the form of long short-term memory networks can be useful in achieving remarkable results in image captioning. Recurrent neural networks have become very widespread in the last few years and we've seen a growing number of attempts to augment RNNs with new properties. Attention can be used so that the model can focus on relevant part of the input sequence as needed as explained in [3]. Sequence-to-sequence models have achieved a lot of success in tasks like machine translation, text summarization, and image captioning.

3. Methodology

The model is a combined LSTM-CNN architecture along with attention mechanism. An LSTM-based Recurrent Neural Network serves as a decoder to translate encoded data into English language descriptions, while a pre-trained Convolutional Neural Network architecture acts as an encoder to extract and encode picture features into a higher dimensional vector space.

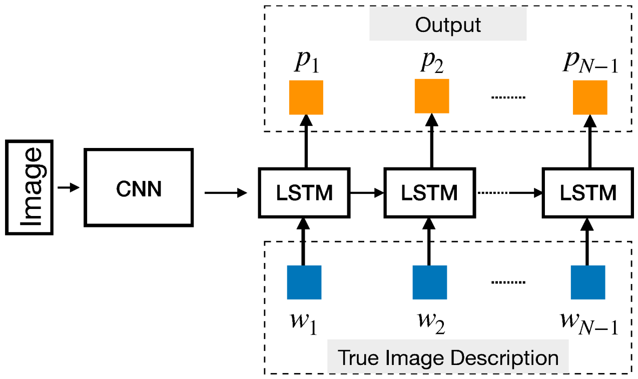


Figure 1: LSTM generating word sequence [4]

3.1. Encoder - Convolutional Neural Network

CNNs are deep, feed-forward ANNs that have been successfully used to analyze visual data. An input layer, an output layer, and numerous hidden layers make up a CNN. Convolutional, pooling, fully linked, and normalizing layers are frequently seen in a CNN's hidden layers. There are numerous other uses for CNN, including natural language processing, recommender systems, and image and video recognition.

CNN transforms the data in the original image into a smaller representation by acting as a feature extractor. This CNN is frequently referred to as the encoder since it compresses the image's content into a smaller feature vector. This phase will result in a feature vector that contains details about the image. Following that, the decoder receives this feature vector as an input. A feature of an image is described by a feature vector, which is a one-dimensional matrix. They are frequently used to describe a feature that is present throughout the entire image or at a specific position inside the image space.

The encoder extracts image features of various sizes and encodes them into vector space which can be later fed to RNN. The most commonly recommended image encoders are VGG-16 and ResNet. I choose to do with ResNet provided by PyTorch library where the hidden size is 2048. This pretrained model is fine-tuned without the last two layers.

3.2. Decoder - Long Short-Term Memory

A fundamental deep learning model that can learn long-term dependencies is the LSTM. A cell, an input gate, an output gate, and a forget gate make up an internal unit of an LSTM. Using simple gating functions, the internal units of LSTM have a hidden state that is enhanced by nonlinear methods to enable the state to propagate without alteration, be updated, or be reset. Handwriting identification, electric load forecasting, and natural language text compression are just a few of the issues that LSTM excel at solving.

The role of the decoder is to interpret the feature vector, or encoded image, as natural language. It would need to be a recurrent neural network because it is generating a series. Every predicted word is used to generate the next word as shown in Figure 1. The right sentence is created using these words with the aid of optimal beam search.

3.3. Attention Mechanism

The attention mechanism uses direct connection to the encoder which allows it to focus on a particular part of the input image. This is done on each step of the decoder. A simple mechanism can be seen in Figure 2. The Attention network computes the weights with respect to significant picture components. To find out important parts of a certain image, all parts of the image are checked out and then we choose what needs describing next.

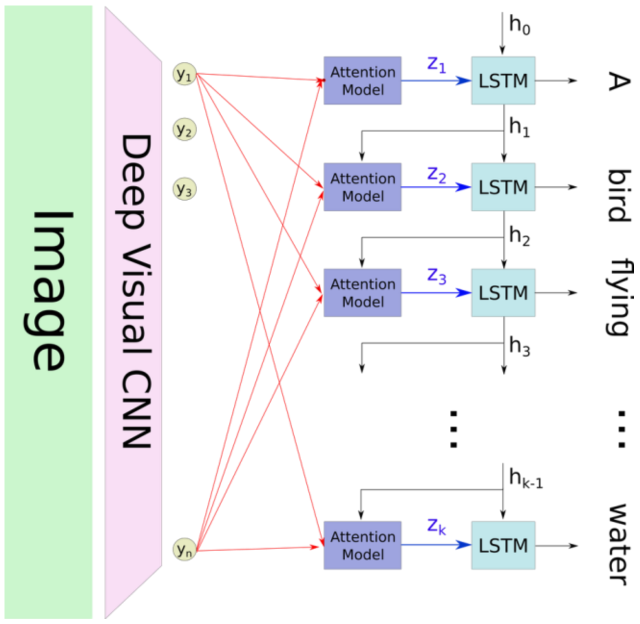


Figure 2: Attention mechanism [6]

The network's attention mechanism instructs it to focus on a certain area of the image in order to produce the subsequent word in the description. By combining encoder output with historical state, which is updated after each iteration, the attention area is calculated. I applied linear transformation to both the encoder output and the historical state output when developing the attention layer. The linear active outputs are summed up and the weight along with attention area is returned. This attention mechanism is trainable with back-propagation and the returned attention area is later fed to the RNN.

3.4. Loss Function

The output of the RNN is a series of likelihoods of word occurrences, so I am using Cross Entropy Loss to measure the output's quality. This is the most common and useful way to evaluate how well a classification model is performing when it produces a probability value between 0 and 1.

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

The above equation is taken from [7], where N is the number of classes, t is either 0 or 1, and p is the predicted possibility that observation k is of class n.

3.5. Beam Search

Beam search is an algorithm used as a final decision making layer to choose the best output given target variables like maximum probability or next output character. It has applications in many NLP and speech recognition models. The beam search expands all viable following steps and keeps the k most likely, where k is a user-specified parameter, as opposed to greedily selecting the most likely next step as the sequence is built. All the k states' successors are formed at each phase. The algorithm stops if any of them is the goal. In any other case, it repeats after choosing the k best successors from the entire list.

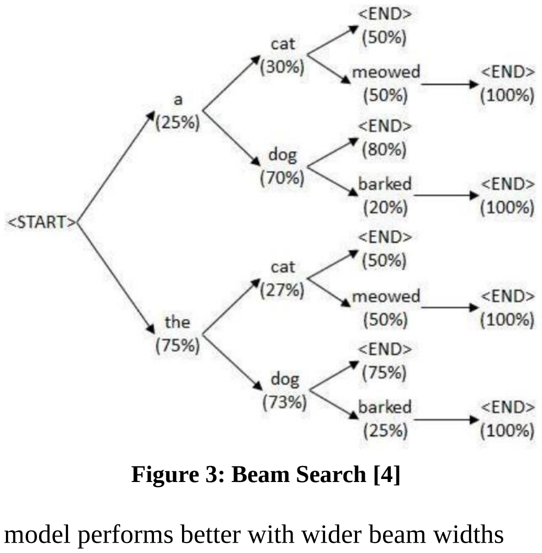


Figure 3: Beam Search [4]

A model performs better with wider beam widths because there are more candidate sequences, which increases the possibility that the candidate sequences will better match the target sequence. Figure 3 shows how beam search works in every step. Decoding speed is slower because of this improved performance. The search can come to a stop for each candidate on its own when it reaches a threshold likelihood, an end-of-sequence token, or the maximum length. The probabilities are small numbers, and very small numbers are produced when small numbers are multiplied together. The probability's natural logarithms are multiplied together to maintain the quantities large and manageable so as not to underflow the floating-point numbers. Additionally, it is frequently done by minimizing the score, thus the negative log of the probability is multiplied. With this last adjustment, it is possible to rank all candidate sequences in ascending order according to their scores and choose the first k as the most likely candidates.

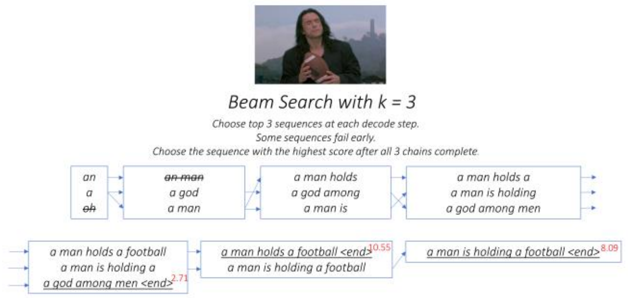

Beam Search with k = 3
Choose top 3 sequences at each decode step.
Some sequences fail early.
Choose the sequence with the highest score after all 3 chains complete.
an
a
~~oh~~
~~an man~~
a god
a man
a man holds
a god among
a man is
a man holds a
a man is holding
a god among men
a man holds a football
a man is holding a
a god among men <end> 2.71
a man holds a football <end> 10.55
a man is holding a football
a man is holding a football <end> 8.09

Figure 4: Beam Search during sentence generation [6]

Beam Search is the final stage in the Show and Tell paper [5] to produce a sentence with the highest likelihood of occurrence given the supplied image [1]. I experimented with beam search sizes ranging from 1 to 5, and I found that 3 tend to produce the best BLEU score evaluations. Thus, I decided to use a beam search with the size of 3.

Figure 4 demonstrates how beam search avoids selecting the word with the highest probability at each step and instead selects the group of words with the highest probability score overall.

4. Experiments

4.1. Dataset

This project uses the Flickr 8K dataset for training the model which can be downloaded from GitHub. It contains 8000 images, and each image is provided with five different captions describing the image in brief. There are 6000 images for training. The vocabulary size for the captions is 7577. Different captions for the same image frequently highlight various features of the scene or employ various linguistic structures. This guarantees that the language used to describe the photos is diverse enough.

4.2. Metrics

BLEU, or the Bilingual Evaluation Understudy, is a score for comparing a candidate translation of text to one or more reference translations. It compares the machine-generated captions to one or several human-written captions and computes a similarity score based on N-gram precision and plus a penalty for too-short system translations. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0. Here, I report my models' performance on BLEU-4.

4.3. Model Performance

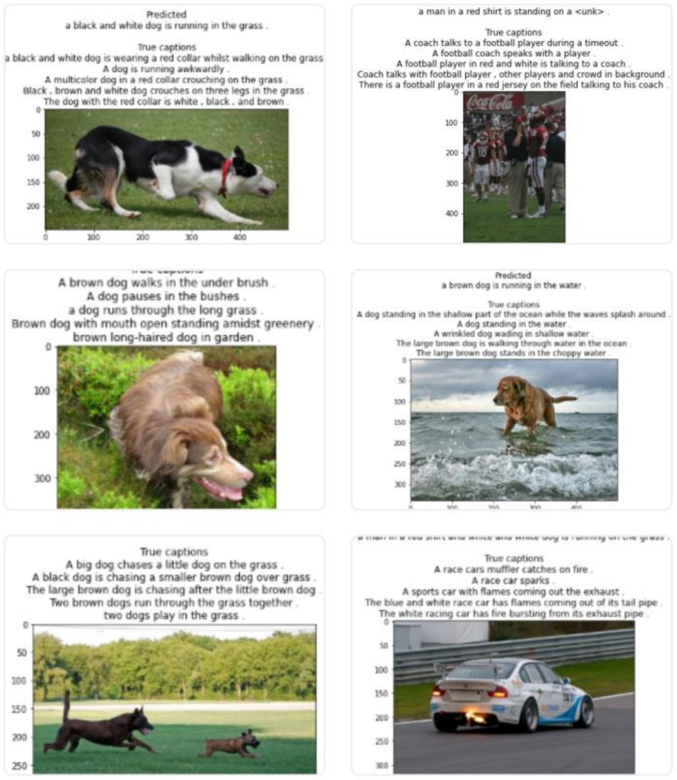
In this study, I altered a few hyperparameters. At first, I changed the batch size to 32 as the given batch size of 5 was too small since the dataset contains thousands of images. Then, I used a learning rate of 3e-5. The teacher forcing ratio is 0. Colab's maximum GPU limit is 12GB, so any batch size over 32 will cause the GPU to go over memory limit. Now the batch size is under GPU limit and big enough for loss to converge.

The pretrained model ResNet is used as CNN after fine tuning for obtaining image features. Since this network won the 2015 ImageNet Challenge, ResNets have become the de facto method for many Computer Vision problems. ResNets demonstrated that Deep Neural Networks of all sizes could be trained without being concerned about the vanishing gradient issue. This effectiveness is due to skip connections, which reduce the vanishing gradient problem by giving the gradient a quicker alternate path to follow. GloVe vectors were used for creating the word embeddings for the captions. The LSTM model has an attention mechanism in badhanau style. The two models are put together in the form of an autoencoder. After putting it all together, I train the model for 20 epochs. It stopped training after 18 epochs because the model was not improving for the last 4 epochs. The best result was achieved at 14 epoch with 11.423% BLEU-4 score and 9.822% validation accuracy. Both training loss and validation loss keep on decreasing. The BLEU-4 score increases rapidly during first 5 epochs and keeps on increasing and decreasing afterwards.

After evaluating the model, I was able to achieve different results for different beam size as seen in Table 1. The best BLEU-4 score is achieved at the beam size of 3.

Table 1

Beam size	BLEU-4 score
1	7.091
3	11.182
5	10.554



Predicted
a black and white dog is running in the grass .

True captions
a black and white dog is wearing a red collar whilst walking on the grass
A dog is running awkwardly .
A multicolor dog in a red collar crouching on the grass .
Black , brown and white dog crouches on three legs in the grass .
The dog with the red collar is white , black , and brown .

a man in a red shirt is standing on a <unk> .

True captions
A coach talks to a football player during a timeout .
A football coach speaks with a player .
A football player in red and white is talking to a coach .
Coach talks with football player , other players and crowd in background .
There is a football player in a red jersey on the field talking to his coach .

True captions
A brown dog walks in the under brush .
A dog pauses in the bushes .
a dog runs through the long grass .
Brown dog with mouth open standing amidst greenery .
brown long-haired dog in garden .

Predicted
a brown dog is running in the water .

True captions
A dog standing in the shallow part of the ocean while the waves splash around .
A dog standing in the water .
A wrinkled dog wading in shallow water .
The large brown dog is walking through water in the ocean .
The large brown dog stands in the choppy water .

True captions
A big dog chases a little dog on the grass .
A black dog is chasing a smaller brown dog over grass .
The large brown dog is chasing after the little brown dog .
Two brown dogs run through the grass together .
two dogs play in the grass .

a man in a red shirt and white and white dog is running on the grass .

True captions
A race cars muffler catches on fire .
A race car sparks .
A sports car with flames coming out the exhaust .
The blue and white race car has flames coming out of its tail pipe .
The white racing car has fire bursting from its exhaust pipe .

Figure 5: Images and Generated Captions

5. Conclusion

The model has been successfully trained to generate captions that are semantically correct with respect to the images. The caption generator has been seemingly improved due to fine tuning of the model with different parameters. Because of the attention mechanism, the model was able to generate more descriptive and rich captions than previous works. By using beam search, the BLEU-4 score has improved significantly. Some sentences generated are not accurate which means even having a higher BLEU-4 score doesn't always give better generated captions. The good and bad captions can be seen in Figure 5. This means that there is room for improvement for better model performance. Automated Image Caption Generator still has a long way to go and can certainly be improved with time given a larger dataset and higher computational power. There are a lot of ongoing research projects which are aiming for accurate image feature extractions and better sentence generation.

6. References

[1] M. Soh, "Learning CNN-LSTM Architectures for Image," Stanford University.

[2] C. Olah and C. Shan, "Attention and Augmented Recurrent Neural Networks," Googele Brain, 2016.

[3] J. Alammar, *Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention).*

[4] N. G. Bhojne, R. Jagtap, R. Jadhav, R. Jadhav and A. Jain, "Image Caption Generation System using Neural Network with Attention Mechanism," *IRJET,* 2020.

[5] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," CoRR, 2014.

[6] S. Vinodababu. [Online]. Available: https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning.

[7] M. Cheatsheet. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html.