DATS 6203: Machine Learning 2
Spring 2022
Instructor: A. Jafari, Ph.D

Image Captioning using Visual Attention in TensorFlow

Group 1: Adel Hassen, Lydia Teinfalt, Xingyu (Alice) Yang
Initials: AH, LT, XY
Last updated: April 26, 2022

1 Abstract

An image caption is a concise sentence describing what is happening in a picture. A good image caption provides context to an image and is useful for retrieving the image at a later point. For the visual impaired population, an image caption is a vital tool for being able to understand a picture without the ability to directly view it. This project demonstrates machine auto-generated captions for pictures in natural language sentence format by implementing in TensorFlow a machine learning model that leverages visual attention mechanism. We use a subset of a large-scale dataset called MS-COCO (Common Objects in Context) to train the model and then generate test captions for New Yorker cartoons in English as well as German. We evaluate the model using BLEU (Bilingual Evaluation Study) score that compares the official caption with the automated generated caption.

2 Introduction

Humans can easily create a caption that describes the content of a picture using a few words. An image caption captures the facts and helps the reader understand what they see in the picture. Image captions are useful not only to people but also to machines. Images with captions are optimized for search engines because they can be indexed by a few words and thus, increasing the accuracy and speed of information retrieval. As machine learning models become more accurate and descriptive in captioning images, they can be used catalog digital archives that are missing descriptions. A task that might take an individual person several years to complete.

Image captions increase accessibility for the visually impaired. The text describing an image is critical to ensure that users with disabilities who cannot see an image can use a tool like a screen reader on a web page to have it read to them instead. Section 508 of the Rehabilitation Act requires that Federal Government must provide U.S. government agencies provide accessibility to electronic information to the public so a machine learning model that can captions for images would be beneficial in staying compliant with the regulation.

This project demonstrates that machine learning models can be trained to generate captions to images. We are implementing a model architecture in TensorFlow based on the paper published by Xu, Kelvin et al called Show, Attend and Tell: Neural Image CaptionGeneration with Visual Attention published in April 2006. The model takes in a raw image uses convolutional neural networks (CNN) to identify objects within an image, translate the image to text, use visual attention mechanism to determine what part of the image the model should focus on and then creates a caption in natural language.

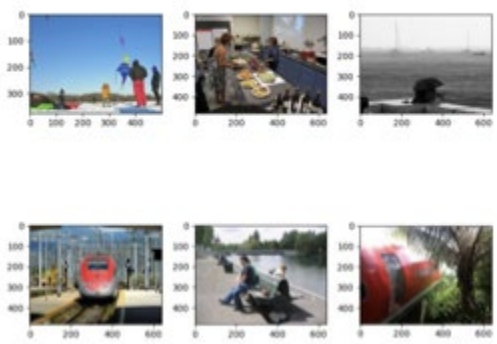
To write a good image caption is a science as well as an art. We use BLEU (Bilingual Evaluation Study) to evaluate captions generated by the model. Instead of measuring accuracy, we can use BLEU score to evaluate quality of machine-generated text against reference text. We compare the results of the model we built against the results reported by the Show, Attend and Tell paper.

The COCO dataset captions are exclusively in English, we added value to the model by translating the captions to a second language. Initially, we were going to investigate using Neural Machine Translation (NMT) as part of our pipeline but with our already slow performing model, we went with using Google Translation API that takes a single sentence at a time and translates it to any of the hundreds of languages supported. Google Translation API would not make sense on a large-scale implementation like creating a French version COCO dataset. As proof of concept, the translation API worked well.

3 Dataset

The MS-COCO (Common Objects in Context) dataset is a classic deep learning dataset that includes both images and their corresponding captions. The dataset was created by Microsoft. It is used in many competitions, and it is also used as a benchmark dataset for state-of-the-art models. This computer vision dataset can be used for image captioning, object detection, image segmentation, and key-point detection. The initial release was in 2014 with around 165,000 images split into training (83,000), validation (41,000), and test (41,000) sets. In 2017, additional images and minor changes were made to the dataset.

We selected the 2014 dataset for our model-building purposes. MS-COCO contains 91 object categories with 82 of the categories containing more than 5,000 labeled images. Compared to the popular ImageNet dataset, MS-COCO has fewer categories but more observations per category which makes it better for computer vision models looking to localize objects in 2D and 3D. The total number of images we used for our project was more than 50,000 images. Each image includes 5 captions resulting in a total of over 250,000 captions. The training set includes 40,018 images and 200,090 captions. The test set includes 10,003 images and 50,015 captions. The data contains an image folder and an annotation folder containing “id”, “image_id”, and “caption”. Below is an example of six images and their captions. One thing to note is that each caption begins with <start> and finishes with <end>.



```
<start> a group of people in a snowy field with kites above <end>
<start> A table with lots of plates of food with wines and a slideshow in the background. <end>
<start> A couple with an umbrella overlooking boats in the water. <end>
<start> a red bullet train is coming towards us <end>
<start> People sitting on a bench near the water.  <end>
<start> A red and white plane in heavy forest area. <end>
```

4 Model

The model uses a Convolutional Neural Network as an encoder to extract features from the input images and feed them into a Long-Short-Term-Memory (LSMT) Network to calculate a context vector, which is then fed into the RNN decoder, generating a predicted word according to the input convolutional feature at each time.
The CNN encoder transforms a raw image data into a 14*14*512 feature map, as the feature map goes through the flatten layer, it is then transformed into a sequence of annotation vectors. Finally, by concatenating the annotation vectors, we get the image feature, which is passed to the LSMT as a input.The LSMT includes a function calculating attention score, a calculation of attention weights as well as a calculation regarding the context vector.

The formula for generating attention score is denoted as

$$score_{t,j} = v_a^T * tanh(U_a * h_{t-1} + W_a * h_j)$$

which contains a hyperbolic tangent transformation of the sum of the multiplication of hidden state ht-1 from the decoder of the previous step with its weight and the output feature map from the CNN encoder with its weight. This score implies how important the jth pixel located in the input image is.

After this, the scores are transferred into an attention weight for the jth pixel at time t with a SoftMax function:

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}.$$

Indicating the probability distribution for each feature map, where the sum of attention weight for all pixels within a single prediction step equal to 1.

Finally, the LSTM network calculates a context vector Ct, by simply summing up the multiplication of each feature and its attention weight calculated in the previous step, formulated as

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j$$

This step combines the extracted features back into an entire image with different weights applied on the blocks. The last state of the model is an RNN decoder. This decoder is a GRU layer followed by two dense layers. It first takes the embedding dimension through an embedding layer and concatenates it with the context vector calculated by the LSTM to get the input batch size, timesteps and features. Then, this concatenated vector is passed to the GRU to get the output and state. As the output went through two dense layers, it can be finally translated into a predicted word.

5 Experiments

We assume that training with 10000 values versus 6000 from COCO 2014 dataset would improve our model. Using the BLEU scores, we saw minimal improvements with the larger dataset. It does not seem to be worthwhile to increase the dataset because it does not significantly improve BLEU scores. The cost of larger dataset slower performance of training the model.

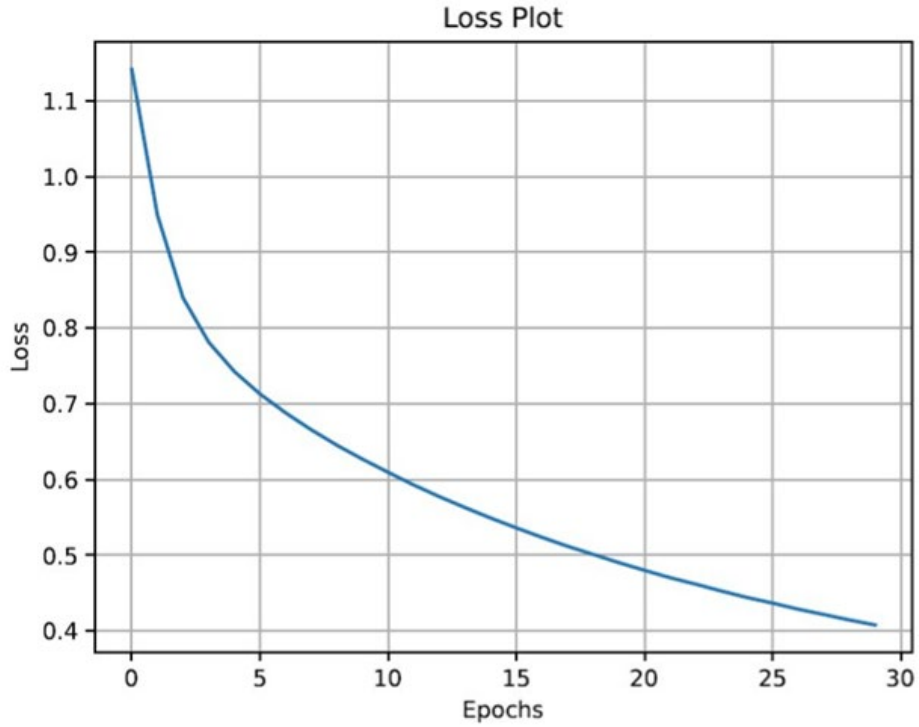
Dataset and Batch Size	BLEU-1	BLEU-2	BLEU-3	BLEU-4
~40,000 Train Images Batch size = 64	44.5	23.3	12.2	6.5
~24,000 Train Images Batch size = 64	44.3	22.0	10.8	5.7
~ 24,000 Train Images Batch size = 128	43.2	21.7	10.6	5.4

We trained the model by varying the number of epochs. The lowest number of epochs being 20 and the highest being 100. By incrementally increasing the number of epochs to train the model, the loss plot shows a steady decrease towards 0. The BLEU score seems to worsen with the increased number of epochs. The best score was a result of training on 10 epochs.

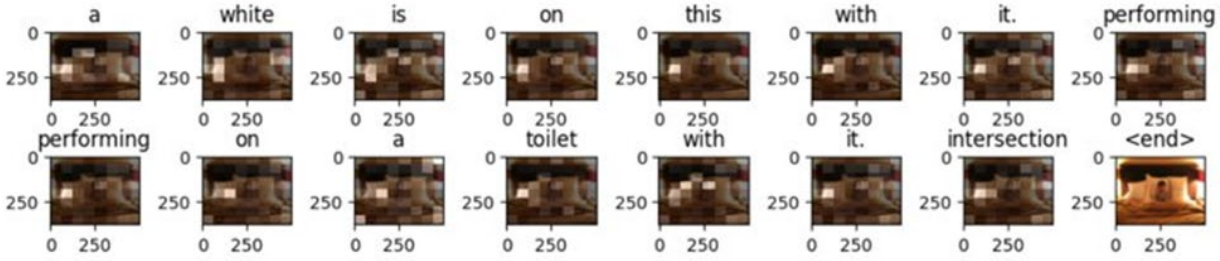
The model's performance is slow to train. On average, an EC2 server on AWS cloud took approximately 160 seconds (about 2 and a half minutes) per epoch to train the model. Assuming that is the average time to train the model, it would take approximately 266.7 hours (about 1 and a half weeks) to train the entire model if we had 100 epochs. The model performed better on a GPU deployed in Google Cloud Platform – approximately 90 seconds (about 1 and a half minutes) per epoch. On a GCP GPU server, it would take us only 150 hours (about 6 and a half days) to train the model if it had 100 epochs.

6 Results

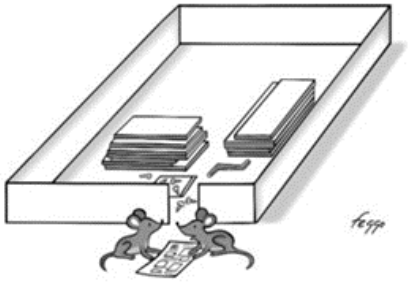
We trained the model on a subset of the COCO 2014 dataset. This resulted in 40,018 training images and 200,090 training captions because there are five captions per image. For the test set, we have 10,003 images and 50,015 captions. We trained using 20, 35, 50, 65, 85, and 100 epochs. Here is the loss plot when we trained using 25 epochs using Adam optimizer. The plot shows a steady decline in loss approaching 0.4.



As part of the evaluation, we ran the model against test and created a single output for validation. The model generated the following caption for the image below which does not make sense in terms of the object identification nor as a sentence.

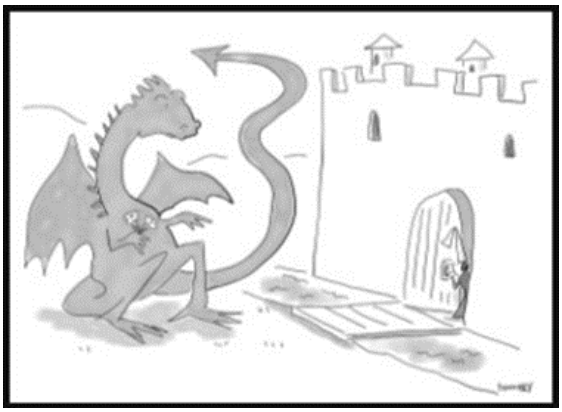


Then we used the trained model to predict captions for New Yorker cartoons. To be fair, we have been training on pictures so to switch to cartoons it has to be acknowledged that cartoons and pictures are not equivalent.



Generated Caption: A table [UNK] a table and a box placed on it. <end>

German Translation: Ein Tisch [UNK] ein Tisch und eine darauf gestellte Kiste. <Ende>
The model correctly identified the rectangular shape as a box but the table detection is not accurate. The two mice were missed by the model entirely. An illustrated mouse is much more an abstraction than a picture of a mouse. The German translation is perfect even if the English sentence does not make sense. Note that [UNK] is for "unknown"



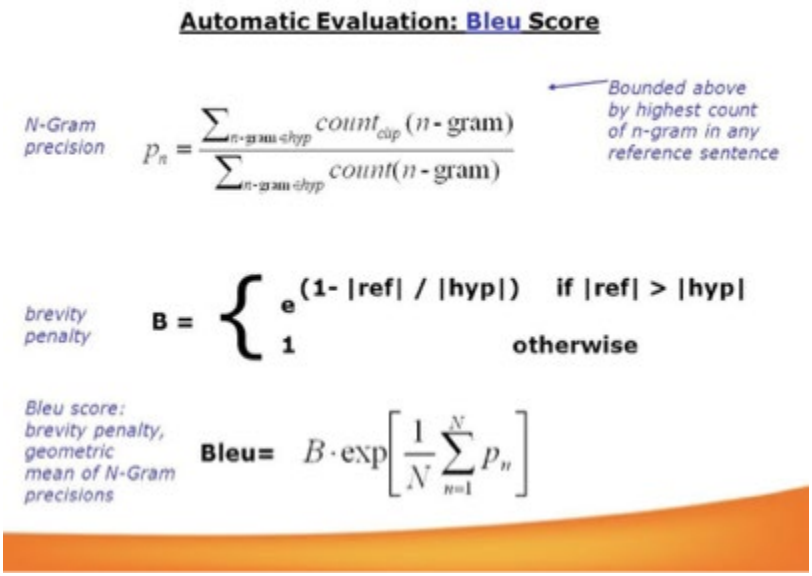
The model identified an animal in the image but incorrectly classified it as a cow. The dragon above also was misclassified as a sheep goat. In terms of the caption the word lounging would make more sense to by the word "lunging" especially following with the word "at." The combination of words sheep goat could be interpreted as the model thinking that a sheep or goat is in the picture. The English caption translated to German is well done except for the sheep goat which is an unknown breed.

BLEU (Bilingual Evaluation Understudy) score is an evaluation metric used to assess the quality of a machine-generated text against reference text (Brownlee, 2019). It was initially developed for translation, but it has been adopted to evaluate other text generating tasks such as image captioning. It ranges from 0 to 1, with 1 being a perfect match of the reference text and 0 being a complete mismatch of the reference. The BLEU score is a simple metric calculation that offers its own strengths and weaknesses. For its strengths, it is inexpensive to calculate, it is widely adopted, and most importantly it correlates highly with human evaluation (Brownlee, 2019). We will be able to understand the flaws of BLEU when we describe how it is calculated.

The BLEU score calculation is simple; count matching n-grams of the candidate text to n-grams in the reference text. Matches are independent of word order. Then, you calculate the precision using the following equation (Khandelwal, 2021).

$$\text{Precision} = \frac{\text{No. of candidate translation words occuring in any reference translation}}{\text{Total no. of words in the candidate translation}}$$

This basic approach falls prey to candidate texts that contain common words like "the" since they will create matches but do not necessarily do a good job of reflecting the reference text (Khandelwal, 2021). Modified n-gram precision clips the counts of n-grams to the number of occurrences in the reference. In addition to clipping, a brevity penalty was also introduced by the authors of BLEU to prevent machine-generated text that is too short compared to the reference text. The brevity penalty and the modified n-gram precision together make up the BLEU score. Cumulative n-gram scores are used in our project; this score is the weighted geometric mean of individual n-gram scores from 1 to n (in our case, n=4). Below is the complete BLEU score formula (Vashee, 2017).



The Show, Attend, and Tell paper includes BLEU scores for several different models. Below, we compare our model to the Soft-attention and Hard-attention models featured in the paper.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Our model	46.4	23.9	12.3	6.3
Soft-attention	70.7	49.2	34.4	24.3
Hard-attention	71.8	50.4	35.7	25.0

7 Summary/Conclusions

We built a working image captioning model in TensorFlow using attention mechanism. Using BLEU score, our model performed poorly compared to the results reported in the Show, Attend, and Tell Paper. The visual attention mechanism is a good discovery that is worth exploring because it resembles what humans do on an intuitive level. There are a lot of areas of improvement for the model. To improve, we could the model using the entire COCO2014 dataset instead of only a subset. In addition, we could parameterize the language selection for translating the image caption so that the output can be in any language that the user specifies. We could have used another image dataset like Flicker8k and created a loop to iterate over the images and generate captions. We could have applied the Hard attention Method to the model since the paper implies that the prediction from hard attention has a higher BLEU score. However, it requires sampling and averaging the result with the Monte Carlo simulation. In contrast, the entire Soft Attention model is differentiable, making it computationally easier to compute the gradient. In this case, we prefer using a soft attention model.

8 References

Brownlee, J. (2019, December 18). *A Gentle Introduction to Calculating the BLEU Score for Text in Python*. Machine Learning Mastery. Retrieved April 26, 2022, from https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

COCO - Common Objects in Context. (n.d.). COCODataset. Retrieved April 11, 2022, from https://cocodataset.org/

Image captioning with visual attention | TensorFlow Core. (2022, January 26). TensorFlow. Retrieved April 15, 2022, from https://www.tensorflow.org/tutorials/text/image_captioning

Section 508 (Federal Electronic and Information Technology). (n.d.). U.S. Access Board. Retrieved April 25, 2022, from https://www.access-board.gov/law/ra.html

Show, Attend And Tell - Paper Explained. (2021, May 3). YouTube Halfling Wizard. Retrieved April 16, 2022, from https://www.youtube.com/watch?v=y1S3Ri7myMg&t=6s

Xu, Ba, Kiros, K. J. R. (2016, April 19). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. Arxiv.Org. Retrieved April 4, 2022, from https://arxiv.org/pdf/1502.03044.pdf?msclkid=97863852c49311ecb15903abb2f078b2

Makarov, Artyom. "Image Captioning with Attention: Part 1." *Medium*, Analytics Vidhya, 14 Dec. 2020, https://medium.com/analytics-vidhya/image-captioning-with-attention-part-1-e8a5f783f6d3.

Sarkar, Subham. "Image Captioning Using Attention Mechanism." *Medium*, The Startup, 15 June 2021, https://medium.com/swlh/image-captioning-using-attention-mechanism-f3d7fc96eb0e.

Khandelwal, R. (2021, December 13). *BLEU — Bilingual Evaluation Understudy - Towards Data Science*. Medium. Retrieved April 26, 2022, from https://towardsdatascience.com/bleu-bilingual-evaluation-understudy-2b4eab9bcfd1

Papineni, Roukos, Ward, Zhu (2022, July 1). BLEU: a Method for Automatic Evaluation of Machine Translation. Dl.acm.org. Retrieved April 26, 2022, from https://aclanthology.org/P02-1040.pdf

Vashee, Kirti. (2017, April 11). *The Problem with BLEU and Neural Machine Translation*. Blogger.Com. Retrieved April 26, 2022, from http://kv-emptypages.blogspot.com/2017/04/the-problem-with-bleu-and-neural.html