Image Captioning

CS 474 Final Project

Ben Murray

The Problem

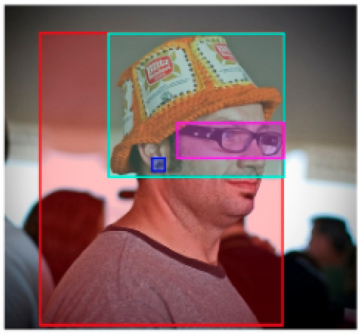
For my final project, I decided to implement a state-of-the-art image captioning system. Image captioning is an interesting problem that has received a lot of attention from researchers. As almost any human can look at an image and use words to describe what's going on, teaching a machine to perform this task brings machine learning one step closer to true general intelligence. It also has a lot of practical applications, like improving accessibility on websites and bootstrapping other computer vision analyses.

Image captioning is a regression problem because the model is generating results that don't fit into a discrete class. I also handled it as a supervised learning problem, using a lot of previously captioned images from a publicly available dataset. My background knowledge made this task a good fit for me. I've worked quite a bit with both natural language processing and computer vision through other classes here at BYU, so I've got a good idea of how feature recognition and word tokenization/vectorization work. A variety of different approaches to this problem have been tried, so I went with one of the more common and well-documented ones, a multi-modal neural network that combines an LSTM based Recurrent Neural Network to generate the words with a Convolutional Neural Network to extract features from the initial image.

The Dataset

Quality data is important in a supervised learning task, and luckily for me, there are several abundant data sources for use with image captioning. The three most famous are Flickr8k, Flickr30k, and the Microsoft Coco dataset. I had already worked with Coco before to do image recognition, so I wanted to do something different for this project. I also had plenty of spare storage and computing time, so I used the larger Flickr30k dataset, found here on Kaggle. Although it's hardly a groundbreaking or novel dataset, it's a standard benchmark for machine learning tasks involving images and associated sentences, with a variety of uses. All of the photos used are obtained from Flickr, the photo sharing website, and the associated captions come from there as well.

The dataset contains 5 captions for each image, as well as coreference chains for those images showing where the different parts of the caption are generated. An example of these coreference chains is shown here. In my models, I did not actually use the coreference chains myself, instead preferring to work with the raw captions and image data. I did, however, implement an attention-based mechanism that does a very similar thing, as will be shown later. It's also worth noting that there's an error with the 2000th image in the dataset that I had to manually fix before running any sort of training or data analysis.



A man with pierced ears is wearing glasses and an orange hat.
A man with glasses is wearing a beer can crotched hat.
A man with gauges and glasses is wearing a Blitz hat.
A man in an orange hat starring at something.
A man wears an orange hat and glasses.

My Approach

Background

After researching for a while, I decided to base my initial image caption model on the approach outlined here, in *Show and Tell*, a landmark paper on image captioning. The model described in the paper is a fairly simple one, feeding directly from a CNN (I opted to use Resnet50) into a LSTM-based RNN to generate the sentences. After working with that for a time, I switched to an improved model based on the paper *Show, Attend, and Tell*, found here. It implements an attention-based mechanism in between the CNN and RNN to focus on different important sections of the image when generating captions.

Initial Model

My initial model was pretty simple. It consisted of two parts, an encoder and a decoder. The encoder was a pretrained, frozen ResNet50 model, with the last layer removed and replaced with my own linear embedding layer. The decoder was a linear embedding layer followed by an LSTM cell, followed by a final linear layer the size of my vocabulary. While this model worked, it wouldn't produce spectacular results, so after a few training loops, the results were worse than I would have liked.

Improved Model

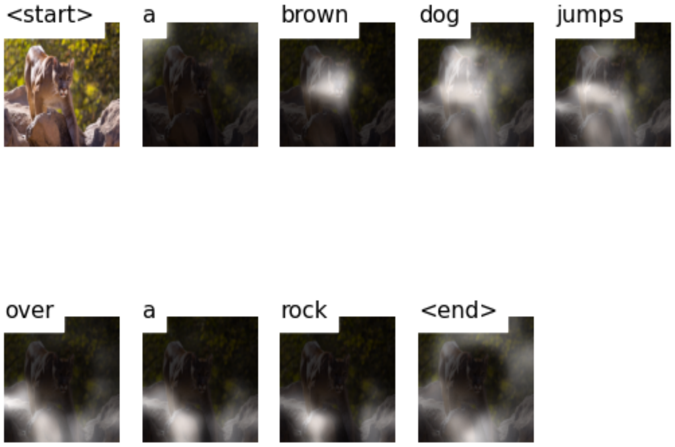
After my initial results proved poor, I moved on to a more advanced model that implemented an attention-based decoder to focus on specific parts of the image. Once again, the model consisted of an encoder and a decoder, but with a series of attention layers in between. The encoder was a pretrained ResNet101 this time, with the last two layers removed and replaced by an adaptive pooling layer to resize the image before sending it to the attention mechanism. The attention model consisted of three linear layers, a ReLU layer, and a softmax layer. It fed into the Decoder, which again, consisted of a few linear layers and an LSTM cell, ending in a sigmoid and one last output linear layer.

Training Algorithm

The training algorithm wasn't overly complex either. I fed the inputs through the model using the Flickr30k dataset with a premade training/validation/test split. Teacher forcing was used to generate the captions, feeding in the first word of the ground truth to the model. I performed validation after every epoch, and logged the BLEU-4 score, one of the primary tools for calculating captioning effectiveness. I stored checkpoints of the model every epoch, and implemented early stopping once the BLEU-4 score stopped improving for more than 5 epochs in a row.

Results

In the end, I trained my model for about 20 epochs before I stopped seeing improvement. It got a final BLEU-4 score of 19.61 on our test set. Human captioners usually score around 21.7 according to the paper above. There's some room for improvement, but that could be solved by training on a larger, more diverse dataset, such as MSCOCO. The model also has the ability to classify any given image, and show the attention mechanism in action. I tried it with several images, including a picture of a cougar as seen here.



All together, I was successful at achieving my goal. I made an image captioning machine that achieved reasonably good results and some really cool output from the attention mechanism. I would consider this project a complete success!

Time Sheet

Date	Time	Task
2/16	.5 hours	Studied various potential projects for ideas
2/17	1 hour	Started to research image captioning - read and understood two papers
2/18	1.5 hours	Decided to work on image captioning - wrote proposal
3/17	2 hours	Looked through available datasets - selected Flickr
3/22	2.5 hours	Configured development environment with cuda and notebook
3/24	2.5 hours	Downloaded and explored dataset - set up torch dataset and debugged some things
4/9	5 hours	Set up models - worked through issues - started training
4/10	3 hours	Switch to Google Colab after training on my machine wasn’t working correctly
4/11	2 hour	Analyze results from initial tests - fix bugs with output generation - decide it's not worth training more
4/12	5 hours	Wrote attention-based model - added visualization tools
4/13	1 hour	Analyzed initial results from attention-based model - decided to go for further training
4/13	2 hours	Started writing final report
4/14	2 hour	Finished model analysis, evaluated on training set
4/14	1 hour	Finished final report
Total:	31 hours	

Image/Dataset Credits

Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik, Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models, IJCV, 123(1):74-93, 2017.

Peter Young, Alice Lai, Micah Hodosh and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions, Transactions of the Association for Computational Linguistics, 2(Feb):67-78, 2014.