Image Caption with Sequence Encoding/Decoding

Boyu Zhang, Mayukh Roy
{bzhang99, rmayukh}@stanford.edu

Stanford University
CS230
Spring 2019

Introduction

Image Captioning is a growing field where using attention model has become a popular approach. Here we explored using an object detection encoder to provide sequential input instead of a traditional CNN, giving us a sequential Encoding and Decoding along with the Attention Model.

Dataset

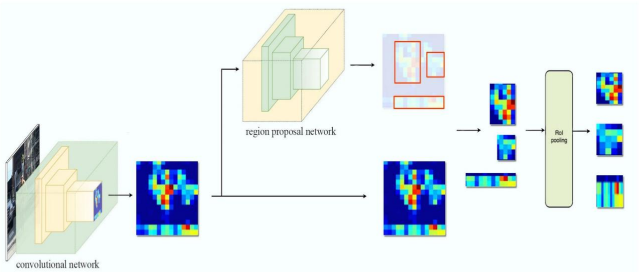
We use COCO (Common Object in Context) as our dataset. In COCO, each image has 5 captions. To save training time, we only use 80k images from COCO training set and split these 80k images into training, dev and test sets

Number of images = 81219
Number of image/caption pairs = 406,095 (81219*5)
Training/Dev/Test set spits = 80/10/10

1. pair of zebras enjoying quiet time in zoo setting.
2. one zebra is laying down and another zebra is standing up.
3. two zebras in a exhibit one standing and one laying down.
4. two zebra standing next to each other on dirt ground.
5. two grown zebra inside some kind of enclosure

Architecture

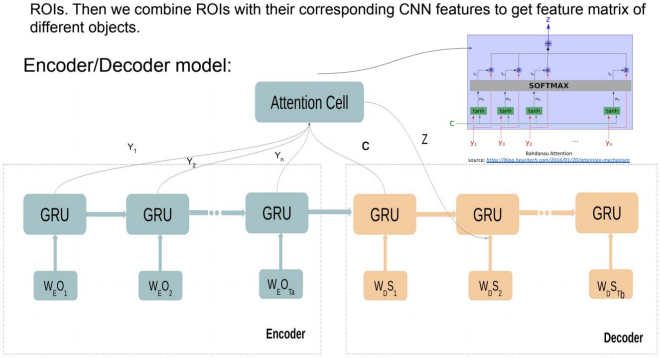
Convert input image to a sequence of objects:



Mask-RCNN model source :https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

We use a Mask-RCNN model to convert the input picture to a sequence of objects. This model first uses CNN to get feature vector of the input image and uses an RPN to determine ROIs. Then we combine ROIs with their corresponding CNN features to get feature matrix of different objects.

Encoder/Decoder model:



Bahdanau Attention
source: https://blog.heuritech.com/2016/01/20/attention-mechanism

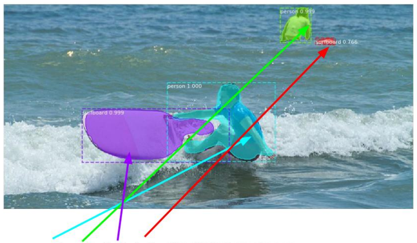
Input sequence: $W_eO_t, t \in \left(1, 2, \ldots, T_A\right)$,$W_e$ = embedding matrix, O_t = CNN feature of t-th object

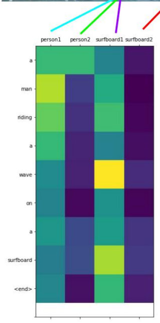
Target sequence: $W_dS_t,\ t \in \left(1, 2, \ldots, T_B\right)$,W_d = embedding matrix, S_t = One-hot vector of input word

Attention mechanism we use is Bahdanau attention

```
score = FC(tanh(FC(encoder_output) + FC(hidden_state)))
attention weights = softmax(score, axis = 1)
context vector = sum(attention weights * encoder_output, axis = 1)
```

Result





Predicted sentence:
Greedy search:
a man riding a wave on a surfboard
Beam search: k = 3
a man riding a surfboard in the ocean
Beam search: k = 8
a man on a surfboard riding a wave

Reference:
1. A young man riding a wave on a boogie board.
2. A man riding on top of a wave with a surfboard.
3. A young man having fun riding a wave to the shore.
4. A man surfs on his surfboard in the water.
5. A boy is falling off of a surfboard in the water.

Attention plot

Model	**Bleu_1**	**Bleu_2**	**Bleu_3**	**Blue_4**
Baseline model with CNN encoder	0.495	0.337	0.221	0.145
Our model	0.683	0.514	0.371	0.264

Future Work

Although the Bleu score of our model is better than the baseline model, it is not as good as other models on COCO leaderboard. In future we can improve our model the following ways:

1. The Mask-RCNN model we use can only detected 80 kinds of objects. We can re-train the Mask-RCNN to detect more objects.
2. Number of objects vary from image to image. We should group images with similar number of objects and train in buckets instead of padding all input sequence to have same length 40.