Image captioning with region attention

Nadir EL MANOUZI
ENS Paris-Saclay
January 20, 2020

Abstract

*The paper Bottom-up and top-down attention for image captioning and visual question answering [1] by Anderson et al. proposed a novel approach in image captioning in a two steps process. First, extracting visual features with a modified version of Faster RCNN and then generating words with an image captioning model which uses attention with these features to focus on salient parts of the image. This approach achieved state-of-the-art on the MSCOCO test server with the framework Caffe but was not evaluated on the Flickr8k dataset. In this paper, I show that this model also improves the state-of-the-art for the BLEU score on this dataset using PyTorch. For this task, I used a pretrained model for the visual features extraction and I trained the image captioning model. The hyperparameters of the last model were tuned to achieve the best score possible. This approach improved the best published results in terms of BLEU-1 from 68.2 to **69.5**, BLEU-2 from 49.6 to **52.6**, BLEU-3 from 35.9 to **38.7** and BLEU-4 from 25.8 to **27.9**.*

1. Introduction

The first direction I took in this project on image captioning with region attention was to implement in PyTorch from the begining the paper Top-down and bottom-up attention for image captioning and visual question answering [1] and train the model with the two big datasets used in the paper (Visual Genome with more than 100k images and MSCOCO with more than 300k images). As I struggled a little at the beginning of the project with this huge datasets, I decided to take another direction.
I took the smallest benchmark dataset used for image captioning (Flickr8k, 8000 images) and worked with it in order to try to show that the model of the paper implemented in two github repositories in PyTorch would also improve the state of the art on this other dataset. Indeed, in the paper, the performance of the model was only evaluated on the MSCOCO dataset. To the best of my knowledge, the evaluation of this model on the Flickr8k was never done. I also did hyperparameter tuning in order to improve the score I got running the model with the author paramaters.
To summarize my contribution, I used a pretrained model to extract the visual features (36 for each image) on the Flickr8k, I preprocessed the data (visual features and captions) for a PyTorch implementation of the captioning model and finally I trained this model and tuned the hyperparameters to achieve the best performance possible.

2. Related work

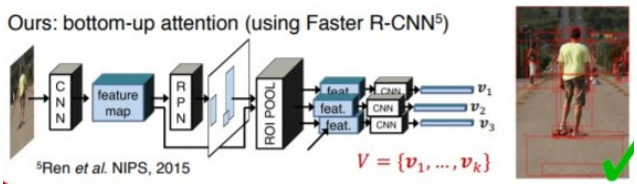
In [2], Chen et al. introduced a novel CNN dubbed SCAN-CNN that incorporates Spatial and Channel-wise attention in a CNN. This model dynamically modulates the sentence generation context in multi-layer feature maps, encoding where and what the visual attention is. In [6], Xu et al. introduced an attention based model and described how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. They also show how we can interpret results by visualizing what the model is focusing on. Finally, in [3], Jia et al. propose an extension of the LSTM model, they add semantic information extracted from the image as extra input to each unit of the LSTM block, with the aim of guiding the model towards solutions that are more tightly coupled to the image content.

3. Approach

Given an image I, the bottom-up attention model outputs a fixed size of image features $V = \{v_1, ..., v_k\}, v_i \in \mathbb{R}^D$ with $k = 36$ and $D = 2048$. Then, the captioning model with a top-down attention mechanism weights each feature during caption generation, using the existing partial output sequence as context.

3.1. Bottom-Up Attention model

The bottom-up attention model is a variation of Faster R-CNN (an object detection model). The model is illustrated in Figure 1. In their implementation, they added an additional training output for predicting attributes classes. The authors used Faster R-CNN with ResNet-101 pretrained on



Source: https://panderson.me/images/cvpr18_UpDown_poster.pdf

Figure 1. Bottom-up attention model

ImageNet and trained their model on Visual Genome data

3.1.1 My work

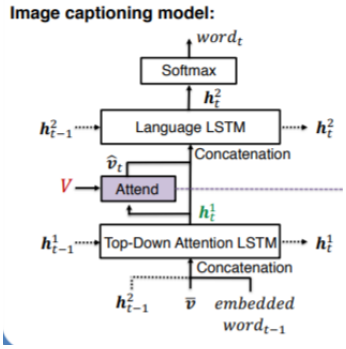
In the author github repository (implementation in Caffe), only bottom-up features for the MSCOCO dataset were available. To obtain the bottom-up features for the Flickr8k dataset on which I worked during this project, I used the github repository of Violetta Shevchenko (https://github.com/violetteshev/bottom-up-features). In this repository, a pretrained bottom-up attention model in PyTorch was available. I modified the code so it can output a fixed size of image features (36) in order to use them with the image captioning model of the next github repository. As output, I had a folder of the same size of the Flickr8k dataset with .npy files. Each file of this folder was associated with an image and contained its 36 features.

3.2. Captioning Model

The captioning model is made of two LSTMs which both take as an input the image features (see Figure 2). The first one, the Top-Down Attention LSTM is taking as one of its input $\bar{v} = \frac{1}{k}\sum_i v_i$ the mean-pooled image feature. Whereas the second one is taking for each time step t as one of its input $\hat{v}_t = \sum_{i=1}^{k} \alpha_{i,t} v_i$, a convex combination of all input features with normalized attention weights.

3.2.1 My work

The second github repository I worked with is https://github.com/poojahira/image-captioning-bottom-up-top-down. The first command needed as input the bottom-up features as a unique tsv file with three field names "image_id", "num_boxes" and "features". In order to run it, I modified the code of a file (generate_tsv.py) in the github repository of the paper author to adapt it to the features in the .npy files and generate the tsv file. Then, I modified the scripts tsv.py and create_input_files to adapt them to my dataset. The first one creates an HDF5 file containing the bottom up image features for train and val splits and PKL files that contain training and validation image IDs mapping to index in the HDF5 dataset. The second one creates JSON files.



Source: https://panderson.me/images/cvpr18_UpDown_poster.pdf

Figure 2. Captioning model

Once all the preprocessing was done, I focused my work on the hyperparameter tuning of this captioning model to achieve the best scores possible on the Flickr8k dataset.

4. Evaluation

4.1. Dataset

I worked during this project on the Flickr8k dataset. One of the three benchmark datasets used in image captioning with the smallest size : 8000 images with 6000 for training, 1000 for validation and 1000 for testing. Each image is associated with 5 reference captions.
I found this dataset more adapted for quick experimentations in this short-term project.

4.2. Quantitative results

In Figure 3, I show the best performance with 4 scores (BLEU-1, BLEU-2, BLEU-3 and BLEU-4) I got with the hyperparameter tuning in comparision with the best published results and other results I found in [2]. The paramaters were the following : 50 epochs, dimension of word embeddings of 1000, number of hidden units in the attention layer of 512, number of hidden units in each LSTM of 1280, batch size of 100, Adamax optimizer and beam size of 6.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Deep VS [4]	57.9	38.3	24.5	16.0
Google NIC [5]	63.0	41.0	27.0	-
Soft-Attention [6]	67.0	44.8	29.9	19.5
Hard-Attention [6]	67.0	45.7	31.4	21.3
emb-gLSTM [3]	64.7	45.9	31.8	21.2
SCA-CNN-ResNet [2]	68.2	49.6	35.9	25.8
Bottom-up Top-down	**69.5**	**52.6**	**38.7**	**27.9**

Figure 3. Final results

In Figure 4 are described the results I got with the hyperparameter tuning. The parameters in the first line cor-

responds to the ones in the implementation of the github repository I worked with. The results are better than the ones I got with the paper parameters. The main difference is the number of hidden units in the attention layer which is two times lower in the paper implementation. The training execution time with these parameters (paper) was also two times lower. Therefore, I decided to keep this number of hidden units in the attention layer (512) and increase the number of hidden units in each LSTM to 1280 and then 1500. The best results were obtained with 1280.

Finally, with this previous best paramaters, I decided to look at the effects of the beam size on the performance. I runned the model with a beam size of 5, 6, 7, 8 and 10. I got the best scores with a beam size of 6 and one can observe that increasing this size doesn't improve necessarily the performance.

4.3. Qualitative results

4.3.1 Image captions

All the following images come from the test dataset.

In Figure 6, the caption generated by the model is "a group of people are sitting on a bench".
The five references are :
1) three people are sitting at an outside picnic bench with an umbrella.
2) three people sit at an outdoor table in front of a building painted like the union jack.
3) three people sit at a picnic table outside of a building painted like a union jack.
4) three people sit at an outdoor cafe.
5) a couple of people sit outdoors at a table with an umbrella and talk.

The model grasps the main information of the image but some details are not described such as the number of people and the fact that this is an outdoor scene.

In Figure 7, the caption generated by the model is "a group of men playing soccer".
The five references are :
1) two boys in green and white uniforms play basketball with two boys in blue and white uniforms
2) four basketball players in action
3) a player from the white and green (unk) team dribbles down court (unk) by a player from the other team
4) young men playing basketball in a competition
5) four men playing basketball two from each team

Here, the model understands soccer instead of basketball. It can be because of the facts that the ball is near the ground and also that a football net appears in the image instead of a basketball hoop which could have helped the model to understand better the scene.

In Figure 8, the caption generated by the model is "a group of people are standing on a sidewalk". The five references are :
1) woman gets her hand (unk) by living statue street artist
2) a woman with a backpack leans (unk) a statue while a group of boys sit on a bench talking
3) a woman is making a statue pretend to kiss her hand beside four boys at a bench
4) a group of tourists stand around as a lady puts her hand near the mouth of a statue
5) a woman posing with a statue alongside a group of boys

In this image, the standing people are described by the model but the main information in the image which is the interaction of a woman with a statue is not captured.

4.3.2 Visualisation of the attention

In order to understand what the model is focusing on during inference, I implemented a code to visualise the attention. I only plotted for each word during generation the bounding box which had the highest attention weight. I give an example of application on an image with two dogs playing in the snow. In Figure 9, I plotted the visualisation of the 36 bounding boxes. And in Figure 10, Figure 11 and Figure 12, one can see the attention of the model for each word generated during inference.

5. Conclusion

During this project on image captioning, I focused my work on a research paper [1] published at a top-quality computer vision conference, adapted available implementations to work on the Flickr8k and showed that this model with hyperparameter tuning improves the best published results in terms of BLEU-1, BLEU-2, BLEU-3 and BLEU-4 on this dataset.

Model	emb_dim	att_dim	dec_dim	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Rouge-L	CIDEr
a (github)	1024	1024	1024	69.1	52.2	38.1	27.2	50.9	68.4
b (paper)	1000	512	1000	67.2	50.1	36.2	25.6	50.6	66.5
c	1000	512	1280	**69.3**	**52.3**	**38.5**	**27.9**	**52.0**	**70.7**
d	1000	512	1500	68.8	51.5	37.4	26.5	50.4	67.9

Figure 4. Hyperparameter tuning with beam size = 5

Beam size	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Rouge-L	CIDEr
5	69.3	52.3	38.5	27.9	52.0	70.7
6	**69.5**	**52.6**	**38.7**	**27.9**	**52.0**	**70.7**
7	69.1	52.2	38.2	27.5	51.8	69.4
8	68.7	51.8	37.8	27.0	51.6	68.7
10	67.9	51.3	37.4	26.7	51.3	67.4

Figure 5. Effects of beam size with emb_dim = 1000, att_dim = 512 and dec_dim = 1280

Figure 6. Prediction : a group of people are sitting on a bench

Figure 7. Prediction : a group of men playing soccer

Figure 8. Prediction : a group of people are standing on a sidewalk

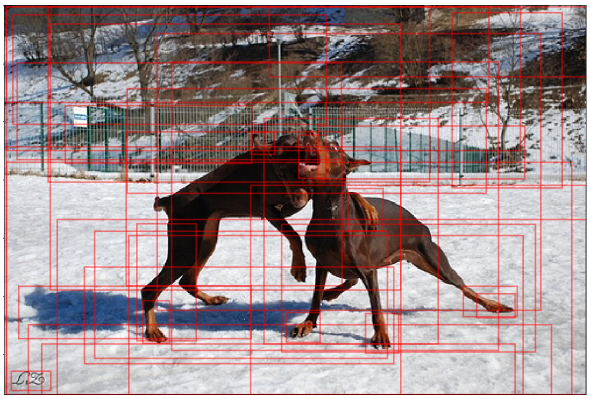


Figure 9. Visualisation of the 36 bounding boxes

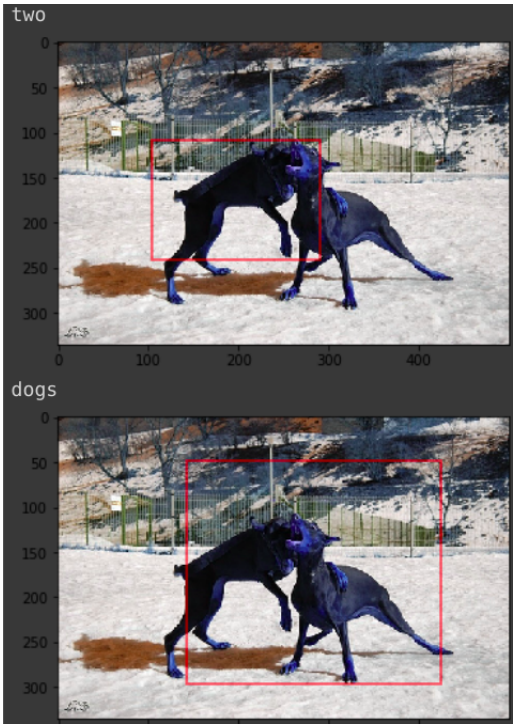


Figure 10. Two dogs ...

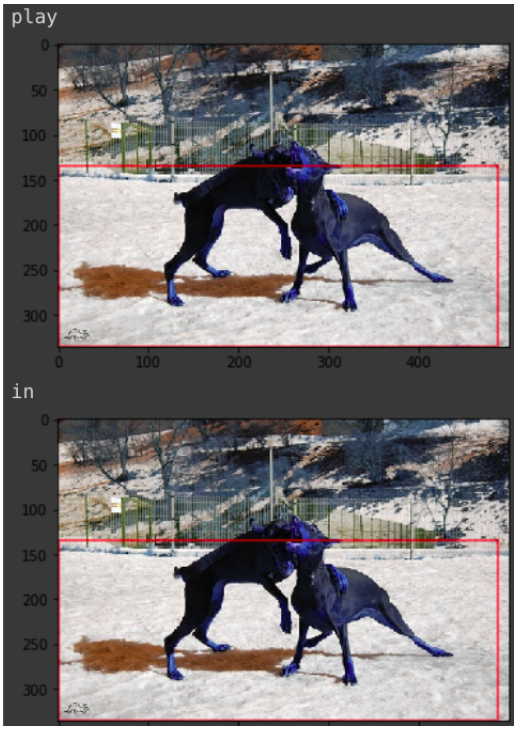


Figure 11. ... play in ...

.

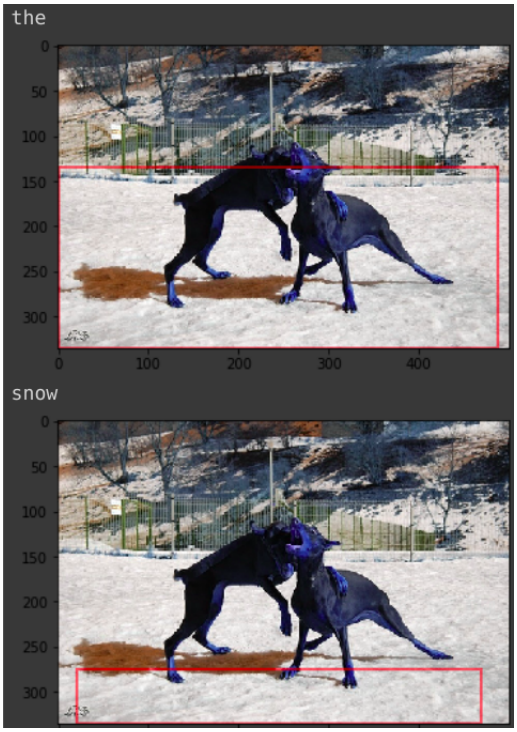


Figure 12. ... the snow

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and VQA. *CoRR*, abs/1707.07998, 2017.

[2] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding long-short term memory for image caption generation. 2015.

[4] Andrej Karpathy and Fei Fei Li. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 2014.

[5] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014.

[6] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015.