**Name- Sanchayan Sarkar Assignment Report Date- 11-11-2020**

*Problem*: To generate captions from images using informative parts of the image.

*Dataset*: For this assignment we use the Localized Narrative annotations [1] which is further built upon multiple Image datasets. For the image dataset, I used Flickr30K entities [2] dataset which has over 30K images with their respective image captions. However, we use the captions from the Localized narrative annotations. Essentially, the annotations are multi-modal in nature where expert annotators not only transcribe image descriptions but also use pointing at the image while they are describing the images. The benefit of this process is two fold : a) we get localized descriptions of images rather than just holistic descriptions and b) it gives researchers a chance to harness the pointing traces which can be aligned with the transcriptions to obtain relationships between transcriptions and their relevant parts in the image. Also, there is speech waveform, so voice analysis and cross-modal studies can also be conducted. In this assignment, I use only the first part of the part where I use only the text transcriptions as for the captioning model.

In this data, an example of a caption would be: “*This image is taken outdoors. At the top left of the image we can see the green grass on the ground. In the middle of the image we can see there are many dry leaves on the ground. We can see the skateboard and a kid is lying on the ground he is with a smiling face.*” Here we can see object entities like <grass>, <ground> <leaves>, <skateboard>, <kid>, <face> and then verb relationships like <lying> and adjective attributes like <green> , <dry>, <smiling>, etc.

*Model:* For creating the captioning model, I followed the model used by [3]. However, the VGG-Net image encoder is changed to ResNet-101 [4] encoder which has proved to be more effective in image classification tasks. Also, I use adaptive pooling on Resnet to obtain a 2D feature embedding. For the decoder part, I use a standard LSTM and an attention mechanism to attend to these image encoding at every time step along with the given caption word at that time step.

For implementation, I adapted the code of [5] for this assignment. The learning rate is 0.0004 for the decoder and the batch size is 128. I also use gradient clipping to prevent exploding gradients and use an early stop criteria based on the BLEU metric which is described below. All punctuations are removed from the transcriptions before using it for decoding. The model is a word-based model rather than a character based model and the output is a probability distribution over the size of the vocabulary. The vocabulary is built using words appearing at least 5 times in the data, a technique also used by the author of the repository.

*Evaluation*: I have used BLEU metric to evaluate the model. BLEU (Bilingual Evaluation Understudy Score) is a metric used for evaluating a generated sentence with referenced sentences (or caption ground truths). A perfect match between two sentences gives 1.0. Although BLEU is done with one or more referenced sentences, in this case, I have used the caption as one continuous sentence. At around the 12th epoch, I got a BLEU score 0.334 with the validation set.

*Discussion and Future Work:* It was interesting problem to work with using two modalities to generate captions. It was my first time combining image and text. Given more time, there are multiple extensions that could be explored with this problem. Firstly, the positional dependencies amongst the sentences within the captions alone could be better captured. Second, the point traces can be used by aligning them with the individual sentences of the caption. That would be a better use of salient features compared to attending over individual sections of the image. Finally, the sound information can also be used for cross modal study.

**Reference:**

[1]. Pont-Tuset, Jordi, et al. "Connecting vision and language with localized narratives." *European Conference on Computer Vision*. Springer, Cham, 2020.

[2] Plummer, Bryan A., et al. "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models." *Proceedings of the IEEE international conference on computer vision*. 2015.

[3] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International conference on machine learning*. 2015.

[4] Wu, Zifeng, Chunhua Shen, and Anton Van Den Hengel. "Wider or deeper: Revisiting the resnet model for visual recognition." *Pattern Recognition* 90 (2019): 119-133.

[5] <https://github.com/Jacklu0831/Image-Captioning>

**Appendix: Steps on how to execute code**

1. Preprocess\_input.py is used to convert the image files to .h5py files and the captions to .json format. (I built this upon the code from the repository. I also preprocessed the text before generating the vocabulary). The directories have to be changed to the respective data directories
2. Train.py – Only use it after the first file. This will train, validate and test and save the model In the output directory. (This is also built upon the code from repository. I have changed the caption.py for data loader while models.py is similar to the repository except when necessary).
3. Generate.py – This is used for testing and getting the best model. This is a new piece of code but the function resembles the validate function of train.py
4. Evaluate.py – This is used for generating captions from just the images. This is also a new piece of code although it has some resemblance with models.py and validate function of train.py
5. Train\_log.txt – Shows the training loss over batches.
6. Model is given inside model directory.

Inline comments are used for better navigation.