Show and Tell: A Neural Image Caption Generator A PyTorch Implementation

Kriti Gupta, Nithish Kaviyan Dhayananda Ganesh, Reshma Lal Jagadheesh, Sarvesh Rajkumar

I. Introduction

THE paper "Show and Tell: A Neural Image Caption Generator" attempts to solve the problem of Automatically describing the content of an image. For problems like Machine Translation, the state of the art solution was to use an RNN with an encoder and which encoded the input to a fixed length vector representation and then have a decoder learn the mapping from that to whatever the required output was. This paper suggests an elegant solution to the problem of image captioning making use of a modified version of the aforementioned technique. This approach solves the issue of being able to describe previously unseen compositions of objects which were observed individually in the training data. The solution suggested here is to make use of a very deep convolutional net to generate the vector representation which is then used as an input to the RNN decoder, which, given the training data, learns the mapping from an image to a sequence of words, which is an 'end to end' network, which is trainable by stochastic gradient descent.

II. Dataset

Flickr8k- Flickr8k consists of 8,108 hand-selected images from Flickr, depiIt consists of 1000 images from 20 object classes, with each class containing 50 objects. The images were taken from 2008 PASCAL development toolkit and were annotated by Turkers who have cleared the qualification test designed by Rashtchian et al. (2010), with each image having five annotations.cting actions and events. Each image consists of 5 captions, annotated by Turkers who have cleared the qualification test designed by Rashtchianetal. (2010)

Flickr30k - This dataset consists of 31,783 photographs of everyday activities, events and scenes harvested from Flickr and 158, 915 captions for these images obtained via crowdsourcing. The Flickr30k dataset is an extension of Flickr8k dataset collected by Hodosh Et. (2013).

COCO - It is a large-scale object detection, segmentation, and captioning dataset. COCO has several features: Object segmentation, Recognition in context, 330K images (greater than 200K labeled), 1.5 million object instances, 80 object categories, 91 stuff categories, 5 captions per image, 250,000 people with keypoints.

TABLE I
Dataset Used

Dataset Name	Train	Validation	Test
Flickr8k	6000	1000	1000
Flickr30k	28000	1000	1000
MSCOCO	82783	40504	40775

III. Model

A neural and probabilistic framework is used to generate descriptions from images. In this project various architectures were implemented for both the encoder and decoder. For Encoder architecture, pretrained ResNets 50, 101 and 152 were used for MSCOCO and Flickr30k. ResNet 34, 50 and 101 were used for Flick8k as the size of the dataset was small. Three combinations of Decoder architectures were used for training the model - LSTM, GRU and Elman. The main objective of the project is to train MSCOCO, Flickr30k and Flickr8k from scratch using the different combinations of Encoder and Decoder. Using the best trained model on MSCOCO and Flickr30k, transfer learning was done on Flick8k.

A. *Evaluation Metrics*

Although it is sometimes not clear whether a description should be deemed successful or not given an image, prior art has proposed several evaluation metrics. The most reliable (but time consuming) evaluation method is Human evaluation. For each of the proposed models, human evaluation was performed by the project team members by choosing 100 images at random and checking the correctness of the description that was given by the model.

Other evaluation metrics that were used for evaluating the correctness of the description were done using automatic methods by comparing the generated caption with the ground truth label. The most commonly used metric so far in the image description literature has been the BLEU score, which is a form of precision of word n-grams between generated and reference sentences.

Besides BLEU, the perplexity of the model for a given transcription is used as another evaluation metric. The perplexity is the geometric mean of the inverse probability for each predicted word. We used this metric to perform choices regarding model selection and hyperparameter tuning in our held-out set. Some of the other metrics such as Meteor and Rouge were also used for evaluation.

B. *Hyper-parameter*

The models were primarily trained using Stochastic Gradient Descent with momentum and using Adam. Perplexity and training loss were the important metrics that were used for deciding the hyper-parameter for the model. Initially, each dataset was trained using both SGD with momentum and Adam for few epochs to check its performance. Even-though SGD without momentum was used in the paper for training the model, we observed that Adam outperformed SGD in both training loss and perplexity. Hence, Adam with a learning rate 0.001 was used for training the model for all the architectures of encoder and decoder.

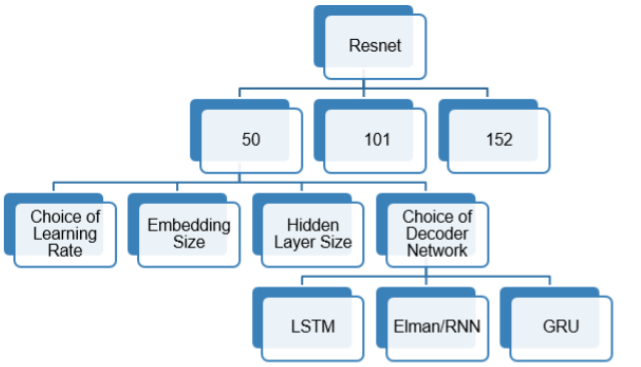

Fig. 1. Different choices of encoder and decoder architecture

We wanted to investigate the effects of depth vs breadth of the neural networks, along with the choice of the decoder network to find out which depth, breadth and decoder combination would yield the best results.

C. *Inference Method*

There are multiple approaches that can be used to generate a sentence given an image. The first one is **Sampling** where we just sample the first word according, then provide the corresponding embedding as input and sample the next word,continuing like this until we sample the special end-of-sentence token or some maximum length. The second method is **Beam Search**. In this project, sampling method is used for generating the captions for the images.

IV. Results

A. *MSCOCO dataset*

TABLE II
MSCOCO TRAIN DATASET

Model	B1	B2	B3	B4	M	R
Resnet 50 LSTM	65.01	45.99	31.65	20.13	19.41	46.81
Resnet 101 LSTM	65.1	45.91	32.64	19.63	18.81	47.08
Resnet 152 LSTM	65.32	46.35	32.89	20.12	19.98	47.71
Resnet 50 GRU	65.13	45.54	31.67	20.02	19.45	46.97
Resnet 101 GRU	65.91	46.12	32.75	19.57	18.65	46.97
Resnet 152 GRU	**66.27**	**48.28**	**33.72**	**22.86**	**21.87**	**48.47**
Resnet 50 Elman	64.87	45.68	31.23	19.85	19.12	45.41
Resnet 101 Elman	64.87	45.96	32.61	19.12	17.54	45.17
Resnet 152 Elman	64.32	46.21	30.13	19.15	18.89	47.08

TABLE III
MSCOCO TEST DATASET

Model	B1	B2	B3	B4	M	R
Vinyals et al.	-	-	-	27.7	23.7	-
Resnet 152 GRU	64.13	46.77	32.98	21.76	20.93	45.91

B - Bleu 1, M - Meteor, R - Rouge

Initially, we expected a model that would not be too deep, with a good breadth level, and with Elman or LSTM would perform the best as these were expected to be less likely to overfit. Our results show that **Resnet 152 GRU** with a breadth of 1024 units performed the best, implying that the deepest, broadest network performed the best, and the decoder used was the **GRU** network, which makes sense because the GRU makes use of relatively more shared parameters than the LSTM.

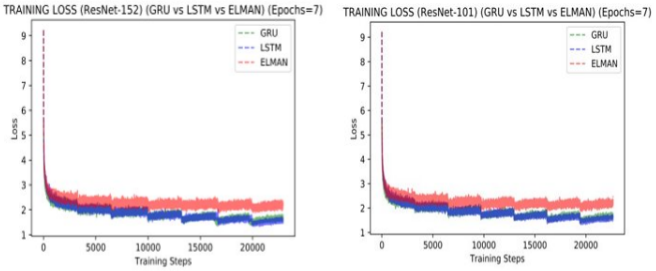

Fig. 2. Training loss results on MSCOCO dataset

From these graphs we can infer that the training loss on the whole is slightly lower for Resnet 152 which is expected because it is a deeper network and hence learns the functional mapping better. Within these two graphs, the trend is relatively similar with Elman tanking out at a higher loss while both LSTM and GRU give similar and lower loss values.

Fig. 3. A selection of evaluation results, grouped by human rating for Coco dataset using Resnet 152 GRU model

*1) **Human Evaluation**:* Based on the quantitative results that were obtained for MSCOCO dataset using evaluation metrics such as Bleu, Meteor and Rouge, human evaluation was done on the best models to check the nature of the caption being generated. 100 images were selected at random and four

human evaluators (Project team members) rated the caption being generated. The average score of all the ratings were taken to choose the best model. Eventhough this is a time consuming way of evaluating the caption, it proved to be a very useful method as it helped us in finding out the nature of the caption being generated by each model. An example of images evaluated by humans on MSCOCO dataset is shown in Fig 3.

TABLE IV
RATING GIVEN BY EVALUATORS IN A SCALE OF 0-9 FOR IMAGE CAPTIONING- MSCOCO DATASET

Model	Kriti	Nithish	Reshma	Sarvesh	Average
Resnet 50 LSTM	5	6	6	6	6
Resnet 101 LSTM	4	5	5	4	4.5
Resnet 152 LSTM	7	6	6	7	6.5
Resnet 50 GRU	3	4	3	4	3.5
Resnet 101 GRU	6	5	5	5	5
Resnet 152 GRU	**6**	**8**	**7**	**7**	**7**
Resnet 50 Elman	5	5	5	4	5
Resnet 101 Elman	6	5	6	6	6
Resnet 152 Elman	7	8	6	7	7

B. *Flickr30k dataset*

TABLE V
FLICKR30K TRAIN DATASET

Model	Enc	Loss	Perp	B1	M	R
Resnet 101 LSTM	**1024**	**1.5825**	**2.7905**	**62.81**	**15.79**	**43.35**
101 LSTM - full	1024	1.0132	1.8105	61.85	14.91	42.62
Resnet 152 LSTM	1024	0.7630	2.1447	62.52	15.01	42.95
Resnet 152 LSTM	512	0.8823	2.4164	61.51	14.73	42.58
Resnet 101 LSTM	512	0.8472	2.3332	61.64	14.98	42.90
Resnet 101 GRU	1024	0.9079	2.4791	61.02	14.25	42.65
Resnet 152 GRU	1024	0.9887	2.6878	61.47	14.83	43.11
Resnet 152 Elman	1024	2.111	8.2576	59.20	12.06	41.83
Resnet 101 Elman	1024	2.2394	9.3880	59.75	13.49	40.69

TABLE VI
FLICKR30K TEST DATASET

Model	Enc	Loss	Perp	B1	M	R
Vinyals et al. -	-	-	-	66	-	-
Resnet 101 LSTM	1024	2.438	11.45	61.39	15.10	42.94

In this our initial assumption that a relatively less deep network with a higher breadth would be able to perform well holds strong - Resnet 101 with 1024 units and an LSTM decoder network(which is more tuneable) outperformed the others. The difference in expected performance between MS COCO and Flickr data can be attributed to the difference between the distribution of images and quality of captions between the two datasets.

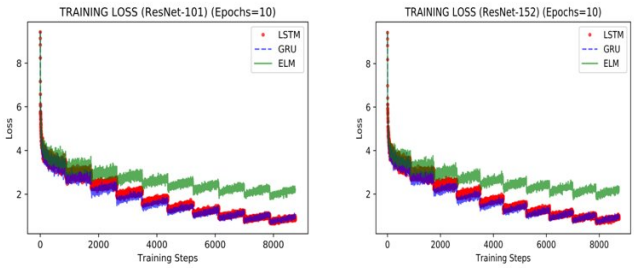

Fig. 4. Training loss results on Flickr30k dataset

The trend in the graphs are similar to the ones we observed on MS COCO data. GRU and LSTM seem to have performed similarly with an overall low training loss with Elman displaying a higher training loss.

C. *Flickr8k dataset*

TABLE VII
FLICKR8K TRAIN DATASET

Model	Enc	Loss	Perp	B1	M	R
Resnet 50 LSTM	**1024**	**1.1383**	**3.1216**	**58.88**	**15.17**	**44.97**
Resnet 50 Elman	512	1.5653	3.5925	49.56	14.98	43.17
Resnet 50 GRU	1024	1.3235	3.3254	54.33	14.86	43.87
Resnet 34 LSTM	1024	1.4625	4.3165	49.57	12.18	42.12
Resnet 34 LSTM	512	1.6192	5.0490	49.75	11.17	42.37
Resnet 34 GRU	1024	1.4565	4.2910	48.22	12.31	43.17
Resnet 34 Elman	1024	2.2199	9.2062	47.78	11.97	42.98
Resnet 50 Elman	1024	2.2256	9.2591	49.25	13.97	43.48
Resnet 101 LSTM	1024	2.1629	8.6965	54.92	14.71	42.94

TABLE VIII
FLICKR8K TEST DATASET

Model	Enc	Loss	Perp	B1	M	R
Vinyals et al.	-	-	-	63	-	-
Resnet 50 LSTM	1024	1.2782	3.1457	57.7	14.82	44.95

In this we notice that the Resnet 50 performed well, this can be attributed to the fact that since this is a small dataset, deeper models tend to overfit and hence do not produce good quality captions. Going by this, Resnet 50 seems to generalise well while paired with an LSTM decoder.

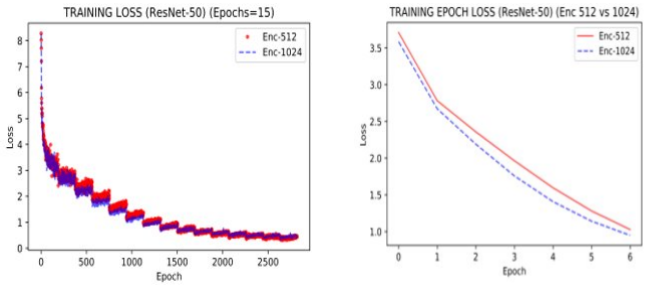

Fig. 5. Training loss results on Flickr8k dataset

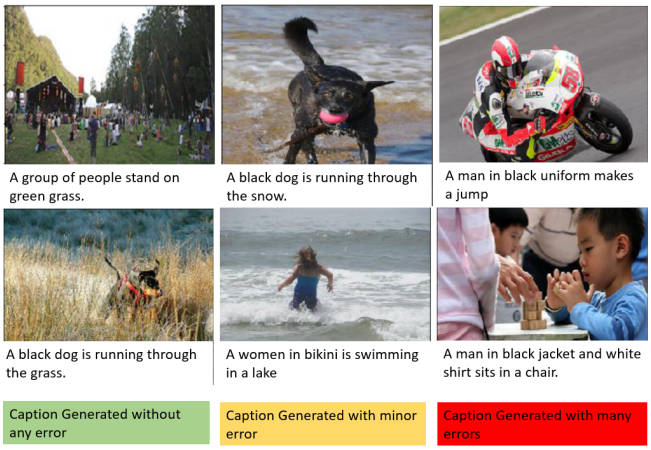

Fig. 6. A selection of evaluation results, grouped by human rating for Flickr dataset using Resnet 50 LSTM model

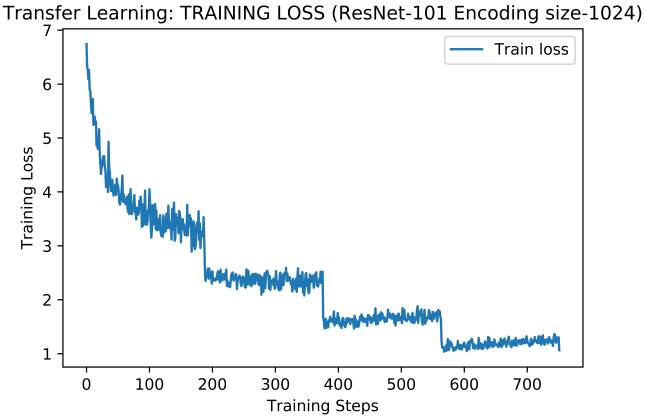

Fig. 7. Transfer Learning- Training loss for Flickr8k when trained using pretrained Resnet-101 LSTM model on Flicker 30k

*1) **Human Evaluation**:* The steps that were followed in MSCOCO for human evaluation was repeated for Flickr dataset as well. Some of the examples of images evaluated by humans on Flickr dataset is shown in Fig 6.

D. Transfer Learning

Transfer learning was done in two ways: one by training on Flickr 8k data using a model trained on Flickr 30k and another by training on Flickr 30k data using a model trained on Flickr 8k. The results are shown in Tables IX and X respectively.

TABLE IX
TRANSFER LEARING ON FLICKR8K USING FLICKR30K WEIGHTS

Model	Train	Test	Perp	B1	M	R
Resnet 101 LSTM	1.063	3.1511	23.11	51.196	12.06	32.98

Table IX shows the result of transfer learning on Flickr 8k with a model pre-trained on FLickr 30k. The metric scores obtained were less than those obtained on Flickr 30k by a model trained on Flickr 30k.

TABLE X
TRANSFER LEARING ON FLICKR30K USING FLICKR8K WEIGHTS

Model	Train	Test	Perp	B1	M	R
Resnet 101 LSTM	2.3787	2.703	10.88	47.87	10.38	30.13

Table X shows the result of transfer learning on Flickr 30k with a model pre-trained on FLickr 8k. The metric scores obtained were less than those obtained on Flickr 8k by a model trained on Flickr 8k. Overall, the results obtained from transfer learning were worse than those obtained from training directly on the corresponding data. Thus, the learning in this case, is data dependant. Also, results obtained from Flickr 30k to 8k were better than those obtained from Flickr 8k to 30k. This could be due to larger size of Flickr 30k data and vocabulary.

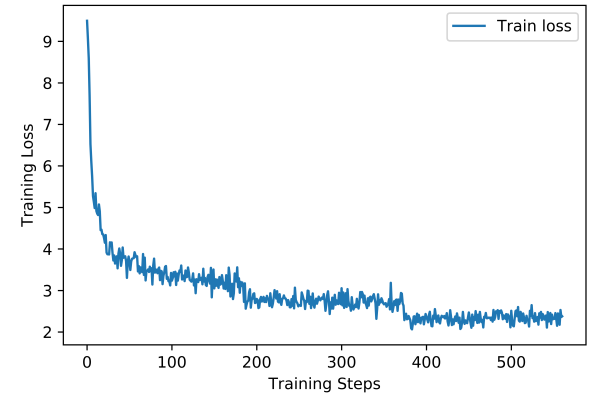

Fig. 8. Transfer Learning- Training loss for Flickr30k when trained using pretrained Resnet-101 LSTM model on Flicker 8k

V. CONCLUSION

This implementation to solve the automatic image captioning problem indicates that this task is highly data dependent i.e; the more data that is available, and with better quality captions, it is able to generalise better. This is validated by our results on transfer learning between Flickr8k and Flickr30k. Although the overall BLEU scores decreased, the BLEU scores obtained from using the Flickr30k embeddings on the Flickr8k data were higher than the other way around. Even while qualitatively evaluated, the captions that were generated by the model that were trained on a larger corpus were evidently better than models trained on smaller corpora. In terms of Model Selection, it was observed that the depth and breadth of the model depended on the size and quality of the dataset. While models with deeper networks and GRU decoders performed better on larger datasets like MS COCO, models with less parameters and were relatively shallower, with LSTMs performed better on smaller datasets(They were able to generalize well). We noticed no significant increase in performance while the word embeddings were initialised with GloVe vectors.

VI. Computational Hours

TABLE XI
TOTAL HOURS CONSUMED

GPU	Hours	Training	Testing
Blue Waters	480	COCO Training and Evaluation	COCO Testing
External GPU – NVIDIA Titan V	250 (which is about 750 Blue Waters hours)	Writing and debugging the code from scratch, Flickr 8/30k Training and Evaluation	Flickr 8k/30k Testing

VII. Acknowledgement

We would like to thank our Professor Justin Sirignano, Teaching Assistants Logan Courtney, Raj Kataria, and Xiaobo Dong for their valuable insights and for their useful suggestions for implementing the paper successfully.

VIII. Future Scope

Training the model using Beam Search Inference as opposed to sampling will yield better results. We were planning on implementing Beam Search Inference but due to time constraints we weren't able to implement it. We were unable to infer results on SBU and Pascal for the same reason, as we did a very wide grid search of the parameters on the first three datasets.

IX. References

1) O. Vinyals, A. Toshev, S. Bengio, D. Erhan. Show and Tell: A Neural Image Caption Generator. arXiv:1411.4555v2.
2) T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. arXiv:1405.0312, 2014.
3) C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 139–147, 2010
4) P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In ACL, 2014.
5) A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In ECCV, 2010.
6) V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In NIPS, 2011.
7) https://en.wikipedia.org/wiki/METEOR
8) https://www.cs.cmu.edu/ alavie/Presentations/MT-Evaluation-MT-Summit-Tutorial-19Sep11.pdf
9) https://arxiv.org/abs/1411.4555
10) http://cocodataset.org/
11) https://machinelearningmastery.com/calculate-bleu-score-for-text-python/
12) https://jair.org/index.php/jair/article/view/10833
13) http://shannon.cs.illinois.edu/ DenotationGraph/data/index.html
14) https://arxiv.org/pdf/1504.00325.pdf
15) https://competitions.codalab.org/competitions/3221
16) https://github.com/muggin/show-and-tell
17) https://github.com/gcunhase/NLPMetrics

X. Legend

Perp : Perplexity
Enc : Encoder
B : Bleu
B1 : Bleu 1
B2 : Bleu 2
B3 : Bleu 3
B4 : Bleu 4
M : Meteor
R : Rouge