

CS 598 DL: Deep Learning Project Report

Adam J. Stewart, Abhinav Kohar, Ke Xu, Shivank Mishra

December 12, 2018

1 Paper Summary

“Show and Tell: A Neural Image Caption Generator” [12] addresses the problem of image captioning. That is, given a query image, it generates complete natural language sentences to describe the image. The model used to solve the problem is called Neural Image Caption (NIC). It first uses a convolutional neural network that has been pretrained on the ImageNet classification problem. The CNN serves as an encoder to encode the query image into a compact vector representation. The image representation is then passed to an RNN network which generates the predicted sequence of words. Specifically, an LSTM network is used to predict each word of the sentence after it has seen the image as well as all preceding words.

The model is trained to maximize the likelihood of the target description sentence given the training image. The description sentence is represented as a sequence of words. Chain rule is applied to model the joint probability as the product of conditional probabilities of each word. Following the above intuition, the sum of the negative log likelihood of the correct word at each step is used as the loss function. The above loss is minimized w.r.t. all the parameters of the LSTM, the top layer of the image embedder CNN and word embeddings.

At the evaluation stage, BeamSearch with a beam width of 20 is used to generate the target sentences. BLEU-1 score is used to evaluate the performance. NIC is able to achieve state-of-the-art performance on various datasets.

2 Datasets

For our project, we used the following datasets. For all datasets that did not have a `Dataset` in `torchvision`, we contributed our own upstream. These datasets will appear in the next release of `torchvision`.

2.1 Pascal VOC

The Pattern Analysis, Statistical Modeling and Computational Learning (PASCAL) Visual Object Classes (VOC) 2012 dataset consists of 11,530 images from 20 classes with annotated objects and segmentations [2]. Since this dataset does not come with captions, we trained

on COCO and used transfer learning to make caption predictions on VOC. Although the `torchvision` library did not come with a VOC dataset, we actively reviewed and helped get <https://github.com/pytorch/vision/pull/663> merged.

2.2 Flickr8k

The Flickr8k dataset consists of 8,000 images, each of which has 5 human-labeled captions [9]. We reserved 1,000 of these images for testing and trained on the rest. We wrote and contributed a dataset for Flickr8k: <https://github.com/pytorch/vision/pull/674>.

2.3 Flickr30k

The Flickr30k dataset consists of 30,000 images, each of which has 5 human-labeled captions [13]. We reserved 1,000 of these images for testing and trained on the rest. We wrote and contributed a dataset for Flickr30k: <https://github.com/pytorch/vision/pull/674>.

2.4 COCO

The Common Objects in COntext (COCO) 2014 dataset consists of 83,000 training images and 41,000 validation images [6]. 4,000 of these validation images are reserved for testing. Luckily, `torchvision` already comes with a built-in dataset for COCO.

2.5 SBU

The Stony Brook University (SBU) captioned photo dataset consists of 1,000,000 images, each of which has a single human-labeled caption [7]. We reserved 1,000 of these images for testing and trained on the rest. We wrote and contributed a dataset for SBU: <https://github.com/pytorch/vision/pull/665>.

3 Model

See Figure 1 for an overview of the entire model.

3.1 CNN Encoder

The CNN encoder takes a query RGB image and encodes it into a compact vector representation. Compared with the original paper which uses GoogLeNet, we use ResNet [3] of different depth as our choice of CNN. ResNet is one of the state-of-the-art CNNs, featuring batch normalization [4] and shortcut connections to speed up convergence and reduce overfitting. The ResNet is pre-trained on ImageNet to harness the representational power and prevent overfitting.

The last fully-connected layer of ResNet is replaced by a linear layer whose number of hidden units is a hyperparameter. Our implementation provides the option to either fine tune the entire ResNet or just the newly added linear layer. A final batch normalization layer is added right after the fully-connected layer.

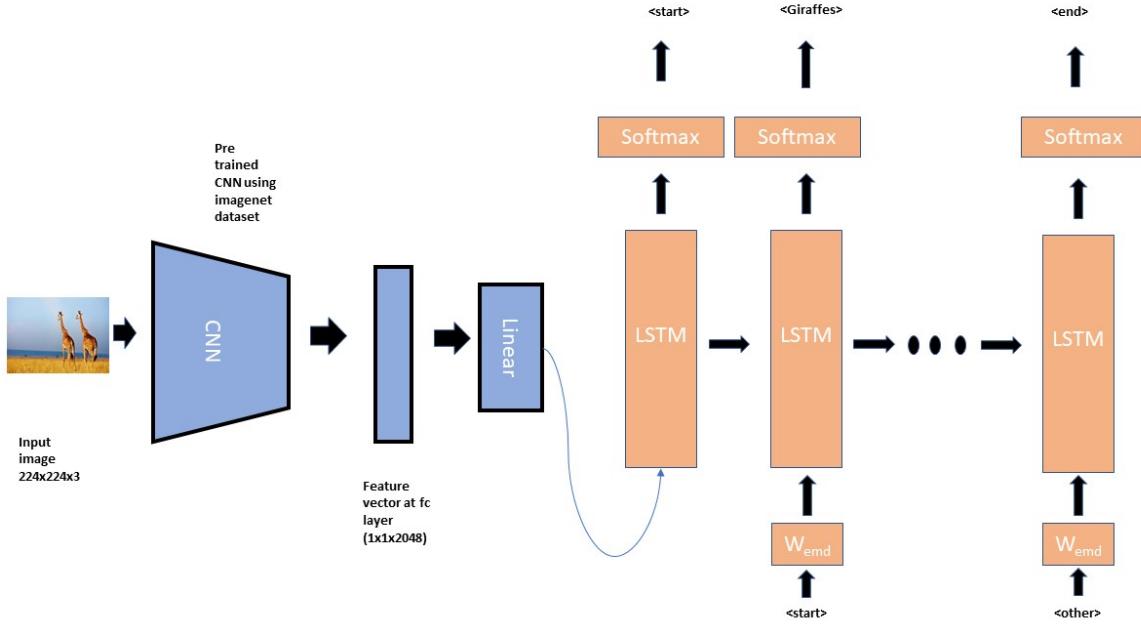


Figure 1: Show and Tell Model

3.2 RNN Decoder

A recurrent neural network is used to generate captions. We pick LSTM as our choice of recurrent unit for our RNN network. The image embedding is only fed into the RNN as a hidden state at the very first time step. The matched captions are fed word by word into the LSTM unit at the following time steps. A linear layer is used right after the LSTM layer to convert hidden embeddings back to vocab.

At each time step, the LSTM unit takes the hidden state and one word as inputs and predicts the next word. The previously generated word input to each decoding LSTM is in the form of a word embedding vector. Word embeddings have a significant advantage over one-hot encodings as they are independent of the size of the dictionary. These embeddings are jointly trained with the rest of the model.

To deal with variable length, the largest sentence vector is chosen and other sentences are padded to make them the same length. For the objective function, we use a masking vector to ignore the padded parts during training.

The forward function of the LSTM class is mainly used for training. It predicts a probability distribution over the vocabulary at every time step. Cross entropy loss is used on the predicted probability as the training objective.

3.3 RNN Sampler

An RNN sampler is used to generate cations for each query image in the evaluation stage. There is no labeled sentence given as input to the LSTM layer, so we use a for-loop to predict one word at each time step and feed predicted word from previous time-steps as

input to current time step. In our final implementation, greedy search is used to pick the best predicted word at each time step.

4 Training

4.1 Training Loss

Below, we show the training loss of the best model on each dataset.

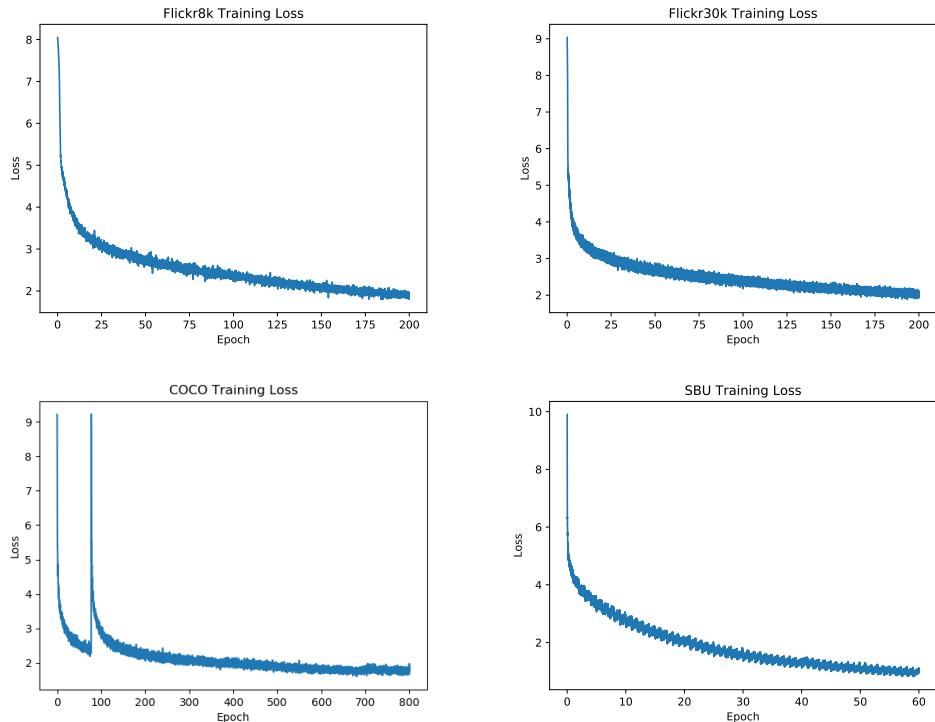


Figure 2: Training loss.

For COCO, we added an L2 regularizer and decreased the learning rate during the 100th epoch, resulting in the spike in training loss you see.

4.2 Hyperparameter Choices

Below we list the best set of hyperparameters for each dataset. Since Pascal VOC does not come with captions, we used the best model trained on COCO.

Dataset	ResNet Size	Embedding Size	Hidden Size	Layers	Optimizer	Learning Rate
Flickr8k	50	512	512	1	Adam	0.0001
Flickr30k	50	512	512	1	Adam	0.0001
COCO	50	512	512	1	Adam	0.001
SBU	50	512	512	1	Adam	0.001

Table 1: Best hyperparameters found for each dataset.

4.3 Computational Hours

All training and testing was performed on Blue Waters. Below are the exact number of hours used to train and test each dataset.

Dataset	Runs	Epochs	Total Hours
Pascal VOC	-	-	1
Flickr8k	18	200	183
Flickr30k	18	200	530
COCO	5	800	620
SBU	3	60	108
Total			1,442

Table 2: Computational hours used for hyperparameter tuning and testing of each dataset. “Runs” is the number of hyperparameter combinations we tried, and “Epochs” is the number of epochs each run was trained for.

5 Results

5.1 Evaluation metrics

We used the following evaluation metrics, using the `nlg-eval` [10] program for calculations.

5.1.1 BLEU

The Bilingual Evaluation Understudy Score [8] (BLEU) is a metric to compare a generated sentence with the reference sentence. BLEU computes the geometric mean of the test corpus’ n-gram precision and adds a brevity-penalty to discourage very short sentences. The perfect match score between candidate, and reference sentence is 1.0 and the mismatch score is 0.0.

5.1.2 METEOR

The Metric for Evaluation of Translation with Explicit ORdering [1] (METEOR) is a metric which computes the harmonic mean of unigram recall and precision. This metric was designed to fix some of the shortcomings in BLEU scores. It utilizes an incremental word alignment method that considers exact word-to-word, word stem, and synonym matches. Recall is calculated at word level. The precision and recall scores are then combined using harmonic mean. The sentences with longer n-gram matches get rewarded.

5.1.3 ROUGE_L

Recall-Oriented Understudy for Gisting Evaluation [5] (ROUGE) is a metric used for evaluating machine translation and automatic summarization . In ROUGE_L, L is the Longest Common Subsequence (LCS) based statistics. It considers similarity and identifies longest co-occurring in sequence n-grams automatically. The advantage of using LCS is that it does not require consecutive matches, but in sequence matches, in-sequence common n-grams, so we do not get an n-gram length that has been decided beforehand.

5.1.4 CIDEr

The Consensus-based Image Description Evaluation (CIDEr) metric performs the sentence similarity by capturing “saliency, importance, grammaticality, and accuracy (precision and recall)” [11]. The n score for n-grams is computed using the average cosine similarity between the candidate sentence and the reference sentences, which account for both precision and recall.

5.2 Quantitative Results

Below we report the results form our model against the paper except Pascal VOC (because it uses human evaluation). For MSCOCO we get better results for BLEU-4 and Meteor, while other metrics are really close, to the ones reported in the paper. For Flickr-8k and Flickr-30k the captions generated are good but we did not beat the paper metrics because we did more hyperparameter tuning, than training a specific model. For SBU, we shot over our computation hours, so we could not train enough, but some captions turned out good. For VOC dataset, the authors used human evaluation, so we have not reported the

Dataset	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE_L	CIDER
Flickr8k	46.76	29.72	18.14	11.11	19.58	33.58	21.38
Flickr30k	55.63	33.99	21.03	13.19	18.18	40.26	29.15
COCO	74.01	56.12	42.13	31.24	26.35	52.91	83.87
SBU	11.74	5.79	3.59	2.67	5.13	14.23	20.54

Table 3: Results of best model on each dataset. Bold numbers indicate better results than the original paper.

Dataset	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE_L	CIDER
Pascal VOC	59	-	-	-	-	-	-
Flickr8k	63	-	-	-	-	-	-
Flickr30k	66	-	-	-	-	-	-
COCO	-	-	-	27.7	23.7	-	85.5
SBU	28	-	-	-	-	-	-

Table 4: Results from the original paper.

5.3 Qualitative Results

For each dataset, we report four example images and their corresponding predicted captions. To find representative examples of both good and bad captions, we sort the results by BLEU-1 score and report an example from four different quartiles.

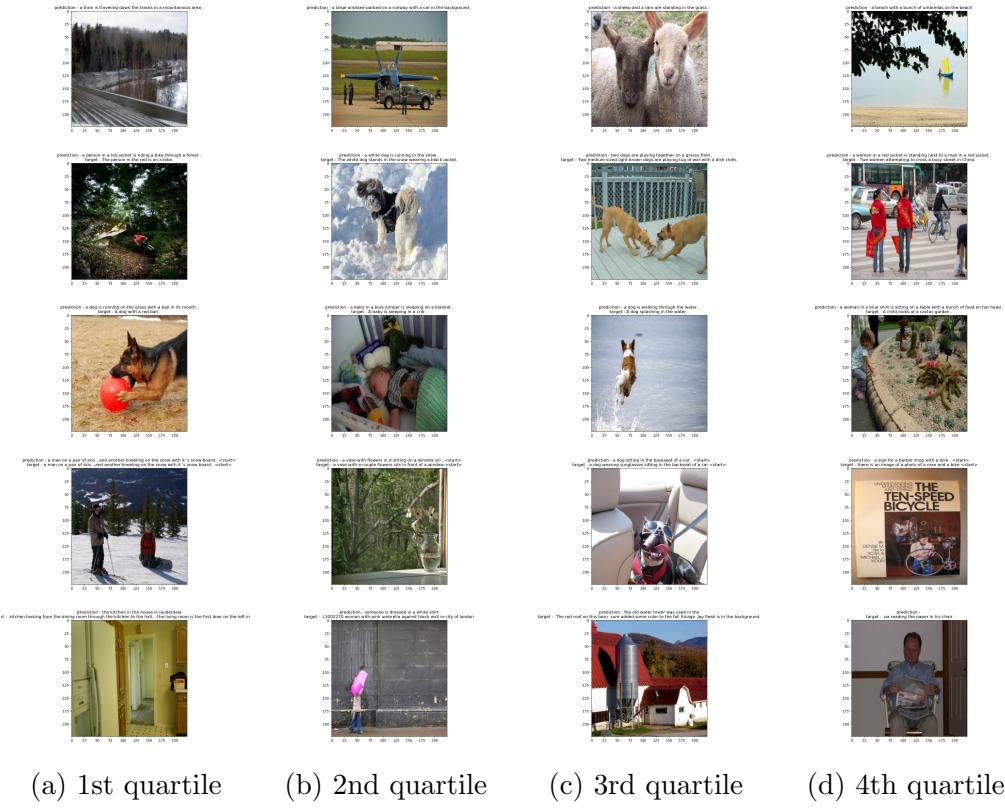


Figure 3: Predicted captions of varying quality. Zoom in to see target and predicted captions. Row 1: Pascal VOC. Row 2: Flickr8k. Row3: Flickr30k. Row 4: COCO. Row 5: SBU.

References

- [1] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *In Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014.
- [2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [5] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.
- [6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [7] Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in neural information processing systems*, pages 1143–1151, 2011.
- [8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.
- [9] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.
- [10] Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799, 2017.
- [11] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.
- [12] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.
- [13] 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*, 2:67–78, 2014.