CS230

Deep Neural Vision for people with Visual Impairment

Niki Agrawal
nikhar@stanford.edu

Mayank Gupta
gmayank@stanford.edu

Akshita Agarwal
akshita@stanford.edu

Abstract

In this paper, we build an Image Captioning system to assist people with visual impairment in outdoor settings. Our models take an image as input and produce a rich, semantic text caption. Our base architecture stems from the Show and Tell model of a CNN encoder, LSTM decoder, and attention network. Techniques like Beam search, Pre-trained word embeddings, and Scheduled sampling were also implemented. Training on blurred images improved the model's robustness without compromising performance on nonblurred images. We evaluate performance through BLEU, other metrics seen in Image Captioning literature, and heat maps.

1 Introduction

Vision is crucial for humans to understand and process the world around us. Modern image captioning systems can greatly aid the visually impaired, by enabling them to take pictures of their surroundings and receive real-time narration of their surroundings. We use a convolutional and recurrent neural network framework to build a system where given an input, the output is a rich semantic text caption of the image. We have optimized for outdoor navigation since these conditions are perhaps one of the most critical scenarios for the visually impaired. We also explored model performance on blurred images. Thus, we performed variants of the MS COCO image captioning task.

2 Related Work

Image Captioning is perhaps one of the most challenging problems of Visual Recognition as it lies at the intersection of Computer Vision and NLP. Several models, such as the Show and Tell model, [16]have explored image captioning on the MS COCO dataset using the Encoder-Decoder architecture with attention, which uses a CNN as an Encoder for Visual feature Extraction and an RNN as a decoder to generate captions. Moreover, Deep Reinforcement Learning [15] techniques like Policy Gradient Optimization [12] and Self Critical Sequence Training [14] are being utilized heavily to reduce the problem of Teacher Forcing and Exposure Bias. RL based algorithms also help the model to optimize metrics like CIDEr score which it will be tested against as opposed to plain Cross Entropy Loss Function. Other works have also explored using more features like Word embeddings [9], Low Level Image features, and various attention models [1] [3] to help the model understand the salient features of an image.

3 Dataset and Features

We extracted images for outdoor navigation from the MSCOCO 2014 [11] dataset using the COCO api by taking all images belonging to the supercategories "Outdoor" and "Vehicles". Then we used the publicly available Karapathy Test split to split these images into the Train, Dev and Test Datasets

,

as shown below. For each image we take 5 captions per image and train on the corresponding Image/Captions pair, example. The Training, Dev and Test set included 32403, 1400 and 1420 images respectively.

Preprocessing We represent each caption as a fixed size tensor. We append <start> and <end> tokens to the beginning and end of each caption respectively. The <start> token is fed as input to the decoder to generate the first word, and the <end> token is used so the decoder can predict the caption end. We encode words appearing less than five times in the set of all captions as <unk>.

Blurring : Note, this section applies only to experiment BLUR. In order to test the robustness of our model on blurred images, we blurred the images in our training, validation, and test datasets using the torchvision Gaussian Blur function with kernel size of 5 and a sigma between 1e-10 (approximately 0) and 6, selected randomly from a uniform distribution. We fixed the blurred datasets to ensure repeatability of the experiment.

4 Methods

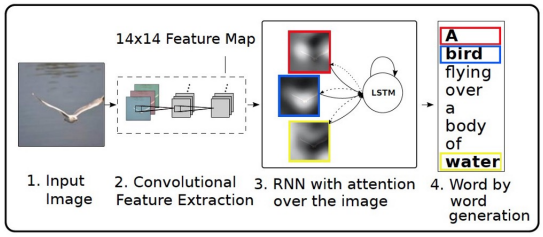

Figure 1: Show and Tell Architecture Diagram

4.0.1 Model Architecture

Our base architecture builds on top of the ShowAndTell[16] paper, which uses an Encoder-Decoder architecture with Attention.

Encoder : We are using the Resnet-101 CNN [8] pretrained on the ImageNet [7] dataset. We are removing the last two fully connected layers as they pertain to the Image Classification task. Resnet-101 produces an encoding of (2048, 14, 14) dimension. This vector represents the visual features of the image and is taken as input by the attention network and to be used by the decoder.

Attention Network: A Soft Attention mechanism has been implemented. It takes as input the encoded image and the previous output of the decoder. The output is an weighted encoded image representation with weights α_i over the pixels in the image. that can be interpreted as probabilities for which part of the image the decoder should attend to.

Decoder : We have used the LSTM architecture for the decoder that converts the encoder input to natural language sentences. At each step, based on the embedding of the previously generated word, hidden state and the output from the Attention Network, a LSTMCell generates a vector of probabilities for each word in the corpus to be present at that position in the output.

Loss Function : We are using the Categorical Cross-Entropy Loss, where each category is a word in the corpus and the decoder outputs a probability for each word to be present at that position in the sequence. Doubly Stochastic Attention regularization ensures our learning algorithm shifts the attention to all pixels over the course of generating the sequence caption rather than just focusing on a few pixels and thus provides a regularization effect. Our loss function is below where L is the number of locations in the image to attend to, C is the length of the caption, t is the timestep, α_{ti} is the weight produced for location i at timestep t by the attention network, and λ is 1.

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2$$

4.1 Techniques Implemented

Beam Search : Beam Search [3] is a greedy tree search algorithm we use at inference time to keep and rank the K best captions generated by the decoder, where K is the beam width, before ultimately selecting the top caption. At each timestep t, we keep the K captions with the highest overall score. After we have K captions that have all terminated, we select the caption with the highest overall score.

Pre-trained Word Embeddings : In some experiments, we load and finetune on top of pre-trained Glove word embeddings for captions. We experimented with both 100D and 300D word embeddings, finding that only 300D improved performance. Due to the limited size of our corpus for outdoor images (about 10K words and 32K training images), we hypothesized that transfer learning in the form of pretrained word embeddings would benefit the model in learning semantic relationships across words for improved captions.

Scheduled Sampling : While training the decoder, we employ Teacher Forcing, which means that at each timestep we pass the *ground truth* word instead of the output of the previous time step of the decoder to generate the next word. This helps the model to converge faster and learn quickly, however it leads to greater variance as we will have to pass the previously generated word from the decoder during testing. With Scheduled Sampling, at each timestep we either pass the previously generated word or ground truth word as input with a sampling probability. In our case, we increase the sampling probability as the model trains for more epochs.

5 Experiments and Results

We implemented ShowAndTell paper with 15K training outdoor images of COCO dataset as our basic model. This model uses Encoder-Decoder Architecture. We tried Beam search of sizes 1, 3, 5 in all experiments and found better performance using it. We increased the dataset size by training on all 32K outdoor images and saw significant increase in performance. We experimented with using pre-trained word embeddings for encoding caption words into the RNN and found 300 dimensional glove word embeddings improves the performance a bit but 100 dimensional embeddings do not perform well. Then we implemented scheduled sampling during sequence training in RNN, to solve exposure bias created by teacher forcing issue in the system. Scheduled sampling further improved the BLEU scores of our system and this turned out to be our best model out of this project.

All implementations are in pytorch and we used 4 cores AWS VM with 1 GPU. we trained all our models using Mini-Batch Gradient Descent and the Adam Optimizer We performed training in 2 steps, first by fixing the encoder weights and just training decoder with 4e-4 initial learning rate and at second step enabling finetuning encoder as well with 1e-4 initial learning rate. This step wise learning helped model converge faster, as in the first step, encoder was fixed, decoder learning became easier. we used adaptive learning rate in all experiments and used to reduce learning rate by 0.8x after every 2 no-progress epochs. To prevent overfitting, We also used early stopping technique. Apart from learning rate, we used following hyper-parameters. These were chosen mainly by literature review and seeing what worked for others. **mini-batch-size**: 80 when encoder was fixed, 32 when encoder finetuning enabled. smaller for finetuning, because it needs more GPU memory for computing gradients in this case and we didnt have enough GPU to fit batch more than 32. **grad-clip**: 5, for gradient exploding problem. **dropout** = 0.5, for regularization effects.

5.1 METRICS

We used Bleu-3, Bleu-4 [13], METEOR [2], ROUGE [10], CIDeR[5] metrics for evaluating our experiment. We kept BLEU as our main optimizing metrics, as it's the most widely used metrics for all NLG tasks. BLEU-N first compute the geometric average of the modified n-gram precisions, p_n, using n-grams up to length N and positive weights w_n summing to one. Next, let c be the length of the candidate translation and r be the effective reference corpus length. Brevity penalty BP and BLEU are calculated as shown in bleu-score.ROUGE is a modification of BLEU that focuses on recall rather than precision and METEOR is similar to BLEU but includes additional steps, like considering synonyms and comparing the stems of words.

Experiment Name	Bleu-3	Bleu-4	METEOR	ROUGE_L	CIDEr
BASELINE	0.364	0.261	0.239	0.504	0.824
BASELINE + BEAM-5	0.393	0.292	0.241	0.516	0.877
FULL-DATASET + BEAM-5	0.416.	0.315	0.246	0.526	0.93
GLOVE-100 + BEAM-5	0.410	0.311	0.251	0.525	0.932
GLOVE-300 + BEAM-5	0.416	0.315	0.252	0.528	0.939
SCHEDULED-SAMPLING + BEAM-3	**0.421**	**0.317**	**0.249**	**0.527**	**0.957**
SCHEDULED-SAMPLING + BEAM-5	0.416	0.314	0.248	0.526	0.936

Table 1: Results of all experiments. BASELINE: ShowAndTell model with 15k images, BEAM-k means beam search was used with beam size of k. FULL-DATASET: uses 34k train outdoor images, GLOVE-n: uses n dimensional GLOVE embeddings, SCHEDULED-SAMPLING: model with Scheduled sampling to reduce variance. Note that SCHEDULED-SAMPLING also uses 300D Glove embeddings.

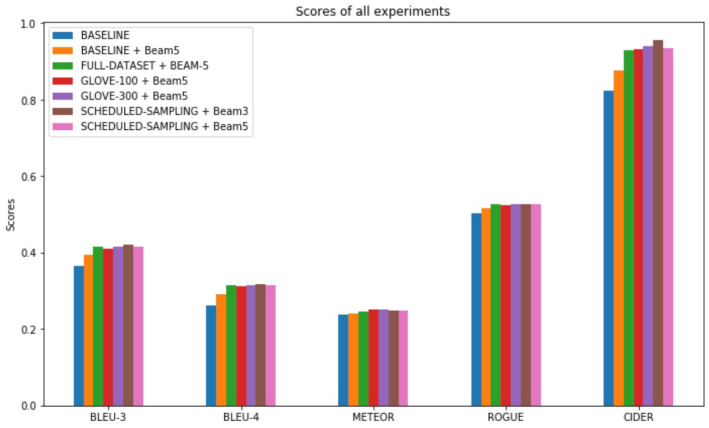

Figure 2: Results graph

5.2 Results

Figure-4 and Table-1 shows performance of the experiments we performed and metrics scores for each one of them. It's clear that Beam search helps a lot and improved Bleu-4 score by more than 10 percent. Increasing the dataset size also improved performance. We also experimented with pretrained Glove word embeddings of size 100 300 and found only 300d embeddings help. Finally scheduled sampling is performing quite well and with beam size of 3 and 300d glove embedding, it is our best model. Interestingly for scheduled sampling model beam size of 5 is performing poorly than 3, while usually higher beam size performs better. There is some existing research done by Cohen and Beck on why this can happen. [6]

Blur Experiment Results : Experiment BLUR uses a blurred training dataset. The goal of these experiments was to improve the model's robustness. On the standard, non-blurred test dataset, the model trained on blurred images shows equal level of performance as the model trained on nonblurred images. However, on blur-test dataset, the model trained with blurring performs significantly better than the model without blurring. This shows that, despite using training datasets of the same size in both experiments, the BLUR experiment increases model robustness without compromising performance on a non-blurred test dataset.

Test Set	Experiment Name	Bleu-3	Bleu-4	METEOR	ROUGE_L	CIDEr
No Blur	**BLUR + PRETRAINED-GLOVE-300 + BEAM-5**	**0.415**	**0.315**	**0.253**	**0.53**	**0.93**
No Blur	**PRETRAINED-GLOVE-300 + BEAM-5**	**0.416**	**0.315**	**0.252**	**0.528**	**0.939**
Blur	**BLUR + PRETRAINED-GLOVE-300 + BEAM-5**	**0.406**	**0.307**	**0.252**	**0.527**	**0.915**
Blur	PRETRAINED-GLOVE-300 + BEAM-5	0.363	0.267	0.230	0.493	0.783

Figure 3: BLUR: blurred training datset, GLOVE-300: uses 300d GLOVE embeddings, BEAM-5 means beam search was used with beam size of 5

5.3 Qualitative Analysis

We analyzed the performance of our best model **SCHEDULED-SAMPLING + BEAM-3** on the Test dataset. The results for good and bad performance can be seen below. Some of the insights we draw from our analysis are:

(a) High BLEU scores (b) Low BLEU scores

Figure 4: Generated Image Captions

1. The model seems to be focusing on detecting the larger objects in the image and reporting them in the captions. This approach works for many cases but is leading to ignoring the smaller objects and salient features of the image. For example in **2f**, it is only reporting "park benches". This is happening as we are only using the last layers of the Resnet which usually contains the more "object-like" features and not the low level ones. Using some features from the middle layers of the encoder can help to alleviate this issue.
2. The model seems to be learning very strong correlation between certain words that appear together in the training dataset, for example, "man" and "skateboard" (as in 1f). Due to this, whenever the model detects one object it learns to generate the other related word in the caption as well(example in **2a**). Such strong correlations can be broken by augumenting the data with images that have Out of Context objects like the OOC dataset [4].
3. The model performs well on images that either have a singular subject or a group of people or objects as in **1a**. However for cases like below where there are 2 subjects in the frame that are not close enough to be called a group, the model prefers to give a singular output. Attention Analysis shows that this is because the attention network has learnt to place the attention on one region as a subject and is unable to shift attention to other object as the primary subject. Modifying the Loss function to penalize for such scenario can help to improve the performance.

6 Conclusion and Future Work

Encoder-decoder + attention based model with scheduled sampling and beam search is the best performing model. Beam search is a heuristic algorithm that can be leveraged in any sequence generation method. State of the art encoder-decoder based methods for sequence generation uses teacher forcing to train the models, which creates exposure bias while training. Scheduled sampling both reduces this exposure bias and leverages benefits of teacher forcing. Our trained models were

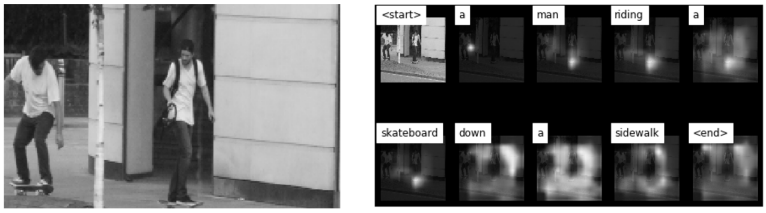

Figure 5: Attention Analysis of an image shows how model gaze is fixed to one entity in a multi-primary-entity image

not robust enough for a test dataset with blurred images. We experimented by adding blurred images in the training data, which alleviated the problem and made our model more robust without sacrificing the accuracy on non-blurred images.

For future work we would like to explore Reinforcement learning based policy gradient algorithms like SCST [14], which directly optimizes the metrics like CIDeR scores. These methods also address exposure bias. Also, other attention models such as Att2In2 and Top Down Bottom Up [1] can be tried to get better performance. Finally, we can apply these techniques to the VizWiz dataset, which is a recently created image captioning dataset with images taken by people with Visual Impairment.

7 Contribution

Each group member contributed to the project equally. The work involved literature review and ideation, setting up the environment for the project, processing the dataset to extract images of interest, implementing and debugging the models, performing Error Analysis and reporting the results. All members collaborated on the proposal, milestone, final report, and video.

References

[1] Peter Anderson et al. "Bottom-Up and Top-Down Attention for Image Captioning and VQA". In: *CoRR* abs/1707.07998 (2017). arXiv: 1707.07998. URL: http://arxiv.org/abs/1707.07998.

[2] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. URL: https://www.aclweb.org/anthology/W05-0909.

[3] Rajarshi Biswas, Michael Barz, and Daniel Sonntag. "Towards Explanatory Interactive Image Captioning Using Top-Down and Bottom-Up Features, Beam Search and Re-ranking". In: *KI - Künstliche Intelligenz* (July 2020), pp. 1–14. DOI: 10.1007/s13218-020-00679-2.

[4] Myung Choi, Antonio Torralba, and Alan Willsky. "Context models and out-of-context objects". In: *Pattern Recognition Letters* 33 (May 2012), pp. 853–862. DOI: 10.1016/j.patrec.2011.12.004.

[5] "CIDEr: Consensus-based Image Description Evaluation". In: (). URL: https://towardsdatascience.com/what-is-teacher-forcing-3da6217fed1c.

[6] Eldan Cohen and Christopher Beck. "Empirical Analysis of Beam Search Performance Degradation in Neural Sequence Models". In: ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, Sept. 2019, pp. 1290–1299. URL: http://proceedings.mlr.press/v97/cohen19a.html.

[7] J. Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[8] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[9] Christopher D. Manning Jeffrey Pennington Richard Socher. "GloVe: Global Vectors for Word Representation". In: (2014). DOI: https://nlp.stanford.edu/pubs/glove.pdf.

[10] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[11] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312.

[12] Siqi Liu et al. "Optimization of image description metrics using policy gradient methods". In: *CoRR* abs/1612.00370 (2016). arXiv: 1612.00370. URL: http://arxiv.org/abs/1612.00370.

[13] Kishore Papineni et al. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: (Oct. 2002). DOI: 10.3115/1073083.1073135.

[14] Steven J. Rennie et al. "Self-critical Sequence Training for Image Captioning". In: *CoRR* abs/1612.00563 (2016). arXiv: 1612.00563. URL: http://arxiv.org/abs/1612.00563.

[15] Haichao Shi et al. "Image Captioning based on Deep Reinforcement Learning". In: *CoRR* abs/1809.04835 (2018). arXiv: 1809.04835. URL: http://arxiv.org/abs/1809.04835.

[16] O. Vinyals et al. "Show and tell: A neural image caption generator". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3156–3164.

8 Appendix

Figure 6: Example of the COCO Image Captioning dataset

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}.$$

Then,

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

Figure 7: BLEU score equation

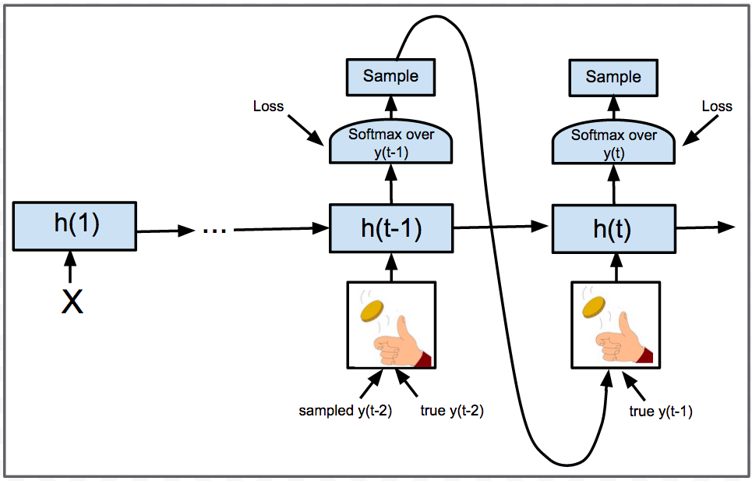

Figure 8: Scheduled Sampling

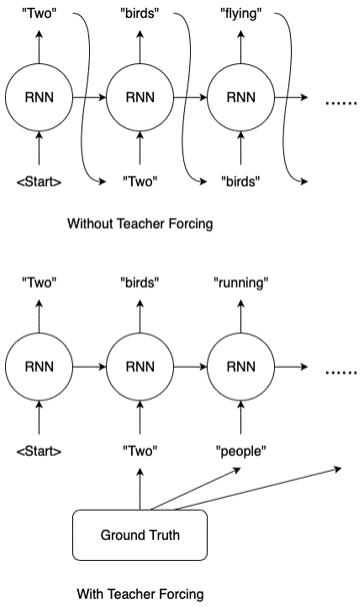

Figure 9: Teacher Forcing