Descriptive Image Captioning using Deep Learning

Ankit Rajvanshi, Deepika Venkatesan, Ratik Vig

New York University
ar7996@nyu.edu, dv2260@nyu.edu, rv2292@nyu.edu

Abstract

One of the most impact full use cases of image captioning is its use in assistive technology. Visually impaired people find it virtually impossible to interact with technology without the help of others. A deep learning model that can recognize subjects and events in an image and describe them clearly can help visually impaired people significantly. People who are visually impaired can interpret image captions by using screen readers, braille displays, talking books, and other similar tools, allowing them to better understand and engage with the visuals. **The code to our codebase is here** link

Introduction

Image Captioning is a process which provides the description of the content within an image. It involves techniques of computer vision and as well as that of natural language processing. Firstly, we need to classify all the objects in the given image. Secondly, we need to generate the description about these objects which express the relation of one object with another along with their respective attributes.

There are many approaches for image captioning but most conventional method is to use encoder-decoder framework which is inspired from neural machine translation. In this approach, input image is encoded into intermediate form of the information contained in the image and subsequently decoded into sequence of text (in natural language like english).

There exists range of CNN architecture based models which can be used as encoder and similarly there are many RNN architecture based models which can be used as decoder. In this project, we explored different models/architecture that can be used as encoder and decoder and finally chose Resnet50 as encoder and GRU as decoder.

Literature Review

Currently, many approaches have been proposed for image captioning. In recent years, a promising research has been emerged due to advances in deep neural network models for Computer Vision (CV) and Natural Language Processing (NLP). In general, there are various types of image captioning modeling techniques, (1) Neural-based approaches, (2) Attention-based strategies and (3) RL-based methods framework. Neural-based and Attention-based approaches are quite popular approaches for image captioning.

In (Kiros, Salakhutdinov, and Zemel 2014), they proposed deep neural networks for image captioning. They suggested to extract characteristics from image to generate captions. More popular and efficient approach was proposed by (Vinyals et al. 2015) where CNN was used for extracting image features and RNN framework was used for generating the language model. They presented an end-to-end network made up of a CNN and a RNN. They trained the model is trained to maximize the likelihood of the target sentence based on the convolutional neural network (CNN) feature of the training image at the starting time step.

In (Karpathy and Fei-Fei 2015), they proposed an image captioning model that combines a convolutional neural network (CNN) and a recurrent neural network (RNN) to process image regions using an alignment model. They employed a bidirectional RNN for language modeling and introduced a structured objective function that aligns two modalities through a multimodal embedding.

The paper (Liu et al. 2020) presents an image captioning model that utilizes generative adversarial networks (GANs) along with retrieval and ensemble-based approaches. The authors draw inspiration from the method proposed by (Deshpande et al. 2019) which employs a variational generative adversarial network and variational auto-encoder to generate image captions based on image summaries. The combined approach aims to improve the quality and accuracy of image captioning by leveraging these techniques.

Many attention based approaches like (Xu et al. 2015) and (You et al. 2016) to image captioning have also been proposed that ground the words in a predicted caption to specific regions in the corresponding image. They learn the distribution of spatial attention during the last convolutional layer of the CNN.

Dataset and Pre-Processing

The dataset that we have used for our project is Flickr30k dataset (Young et al. 2014) (Plummer et al. 2017). This dataset is a benchmark for image descriptions. It contains 31,000 images each containing 5 reference captions added by human annotators.

Figure 1: Sample images from Flickr30k dataset (Young et al. 2014)

Dataset Split

We used a binomial distribution with p=(0.1) to split our dataset into a training and validation set. We get 28,605 images in the training set and 3,871 images in our validation set.

Building Vocabulary

We use the annotation file from the dataset to build our vocabulary. We tokenize all the captions from the annotation file, transform them to lowercase and add these to the vocabulary. We also add some special tokens to our vocabulary like ⟨start⟩that donates the start of sentences, ⟨end⟩denoting end of sentence, ⟨pad⟩to pad smaller sentences making sure all sentences are of the same length and ⟨unk⟩for predicted words that are not in the vocabulary.

Data Augmentation

To further boost our predictions, we use data augmentation so that our model can train of a bigger set of images. While creating our data loader, we use transforms such as crop, horizontal flip and vertical flip. We also normalize our images by calculating the mean of all the images in the dataset.

Model

We have used an Encoder-Decoder architecture for our project. The Encoder is a pretrained Resnet-50 (He et al. 2016) model as shown in figure 3, and the decoder is a one hidden layer GRU (Chung et al. 2014) for caption generation as shown in figure 4 and 5 . We do not update the weights of the Resnet model, but at the last layer, we added a batch normalization layer and a fully connected layer. The Encoder is

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)}$$

Figure 2: Cross Entropy Loss

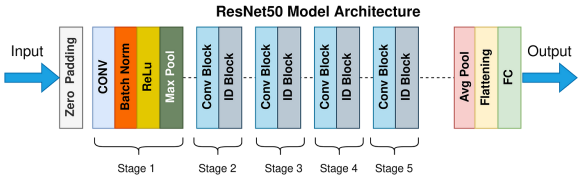


Figure 3: ResNet50 Architecture

used to extract relevant features from the images into a feature vector of size 512. The features that the encoder extracts are then passed on to the decoder, in our case, the GRU. The first layer of the encoder is an embedding layer of size (vocab_size, 512). The output from the word embeddings is then passed on to our hidden layer of size (512, 1024). The weights of the GRU (embedding layer and the hidden layer) are then trained over several epochs. We compare the captions from the dataset to our generated captions using a loss function. The loss function that we have used is a Cross Entropy Loss as shown in 2 where $\hat{y}$ is predicted output and y is expected output and K is length of the output layer of GRU.

Experiment

Evaluation Metrics We are using state-of-the-art BLeU score to evaluate the performance of our model. As per (Microsoft 2022), "The BLEU algorithm compares consecutive phrases of the automatic translation with the consecutive phrases it finds in the reference translation, and counts the number of matches, in a weighted fashion."

The basic idea involves computing the precision – which is the fraction of candidate words in reference.

During computation it takes into account individual words or unigrams of candidate that occur in target/reference. However, for a more accurate evaluation of the match, one could compute bi-grams or even tri-grams and average the score obtained from various n-grams to compute the overall BLEU score.

We used 2-gram overlap to compute the score. We got max validation BLeU score of 0.404 on the scale of (0,1) while using Resnet-50 as encoder, LSTM as decoder with Adam as optimizer with 0.01 as learning rate. While with our final model where GRU was used as decoder, we got max validation BLeU score of 0.350.

In case of Testing, we got BLeU score of 0.259 with GRU and 0.197 with LSTM.

Optimizer Initially, we trained our model with Adam Optimizer with 0.001 learning rate and got the max BLeU score of 0.385 and training loss was reduced to 1.8580 in 10 epochs. Later, we tried the using RMSprop with 0.001

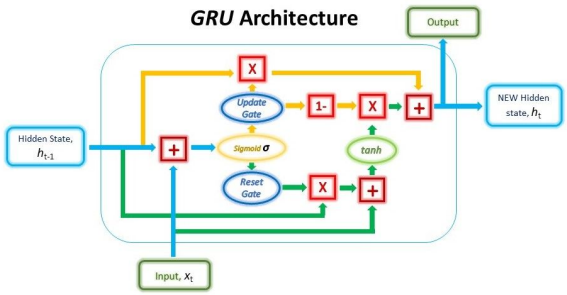


Figure 4: GRU Architecture

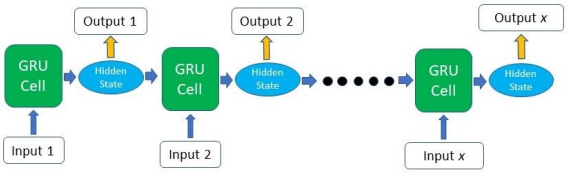


Figure 5: GRU Flow

learning rate and we got max BLeU score of 0.399 while training loss was reduced to 1.7805 in 10 epochs.

Hyperparameters We tried to train model with different learning rates, different embedding sizes and hidden sizes. We trained LSTM model with 0.001 and 0.01 learning rates. We ran some models with embedding size of 256 and hidden layer size of 512 and others with embedding size of 512 and hidden layer size of 1024. By increasing the size of the embedding layer we would be able to capture more features that were extracted by the resnet model.

We also tried to change number of layers in decoder but need to drop this idea as it was taking more epochs to give better results.

Results

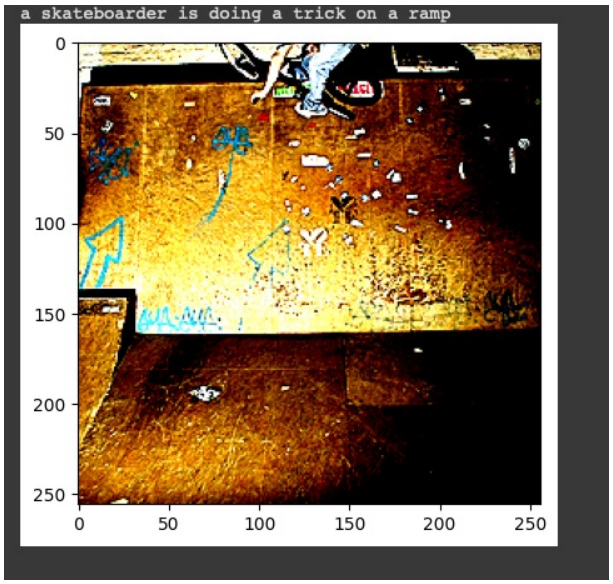
We were able to generate very good captions with Resnet-50 as encoder and GRU as decoder as shown in figure 6.

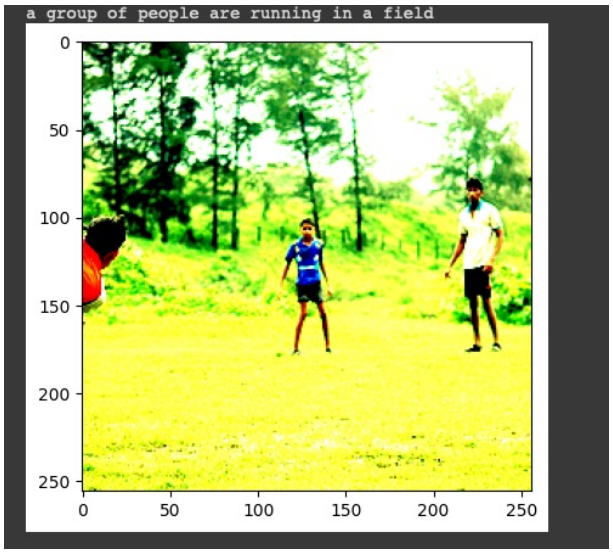
Table 1 shows BLeU scores for various configurations of model we have tried.

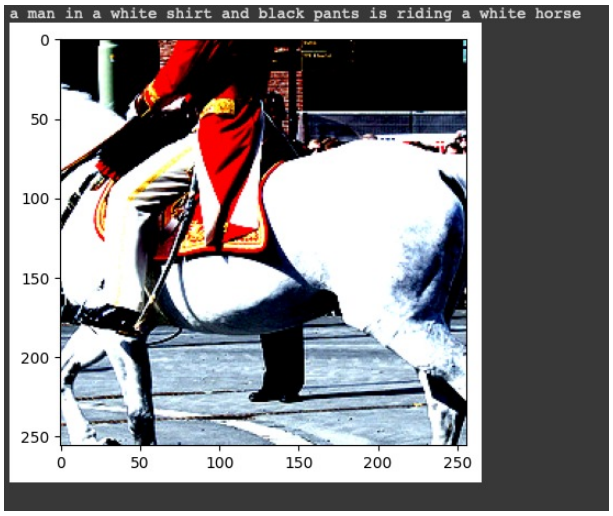
Figure 7 shows Training loss over 10 epochs for our final model.

Table 1: Different model configuration with BLeU Score

Decoder	***Optimizer***	***Val BLeU Score***	***Test BLeU Score***
LSTM	Adam	0.404	0.197
LSTM	RMSprop	0.399	0.125
GRU	Adam	0.350	0.259







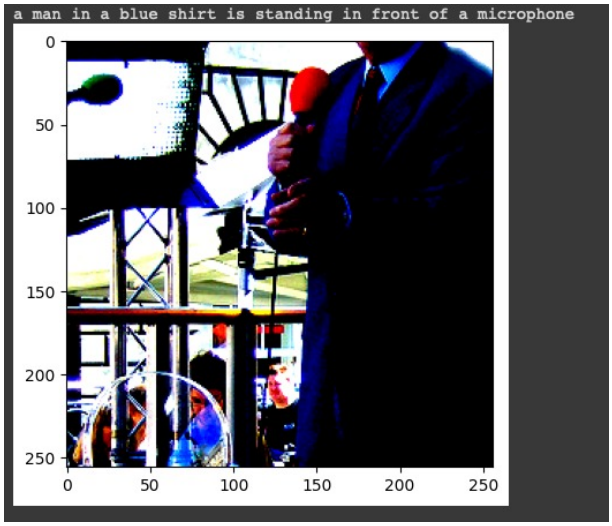


Figure 6: Generated Image Captions

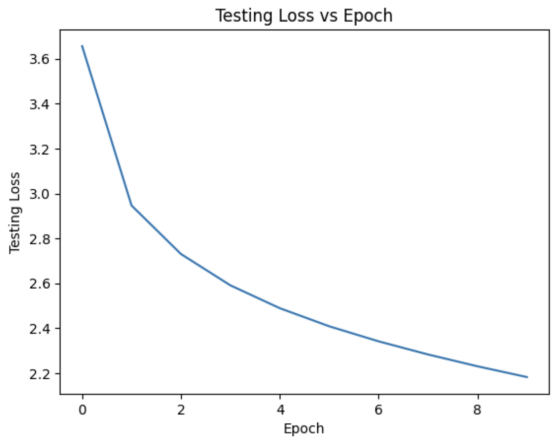


Figure 7: Loss vs Epoch Plot

References

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Deshpande, A.; Aneja, J.; Wang, L.; Schwing, A. G.; and Forsyth, D. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10695–10704.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Karpathy, A.; and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137.

Kiros, R.; Salakhutdinov, R.; and Zemel, R. 2014. Multimodal neural language models. In *International conference on machine learning*, 595–603. PMLR.

Liu, J.; Wang, K.; Xu, C.; Zhao, Z.; Xu, R.; Shen, Y.; and Yang, M. 2020. Interactive dual generative adversarial networks for image captioning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11588–11595.

Microsoft. 2022. What is a BLEU score? https://learn.microsoft.com/en-us/azure/cognitive-services/translator/custom-translator/concepts/bleu-score.

Plummer, B. A.; Wang, L.; Cervantes, C. M.; Caicedo, J. C.; Hockenmaier, J.; and Lazebnik, S. 2017. Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. *IJCV*, 123(1): 74–93.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057. PMLR.

You, Q.; Jin, H.; Wang, Z.; Fang, C.; and Luo, J. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4651–4659.

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2: 67–78.