Image Caption Generator

Basumitra Chaki **Srinivas Jakka** **Taehoon Kang**
School of Information, MIDS program
University of California at Berkeley
Berkeley, CA 94720
{ basumitra, jakkas, taehoonkang }@ischool.berkeley.edu

Abstract

Automatic caption generation is gaining attention in the field of Artificial Intelligence (AI) with rapidly developing machine learning technologies. In this paper, we followed prior works that utilized the encoder-decoder framework in which we used pre trained CNN models, VGG16 and Xception, as image parts. The LSTM model is utilized for the language decoder part. The baseline model performance measured by the BLEU score was limited. To improve performance, we introduced the attention mechanism and which outperformed the previous CNN + LSTM models. The hyperparameter tuning boosted the transformer model performance and achieved 32.9% BLEU score.

1 Introduction

AI is more powerful than ever with recent advances in deep learning. Successes in image processing and the natural language processing (NLP) fields made AI driven image caption generation comparable with what humans perceive. However, when we think about the task of describing an image from a machine's perspective, it is still difficult to build a model that creates optimal image captions with semantically and syntactically proper language. It requires the capability to identify objects in an image and, at the same time, understanding of NLP to narrate the relations and attributes of identified objects.

Extensive research has been conducted in the field of image captioning and models have changed over time, but the recent approach was to use Convolutional Neural Networks (CNN) for the encoder and the Long Term Short Term memory (LSTM) network for the decoder. We replicated the models using pre-trained encoder models such as Xception, and VGG-16 which are pre-trained on 1.2 million and 14 million images respectively. Preliminary results using these models were disappointing therefore we introduced the attention mechanism to enhance the model performance.

We used the Flickr 8k dataset that Brownlee (et al) [1] referred to in their research article. This dataset contains 8091 images and 5 corresponding captions for each image. Bigger datasets were also available such as the MSCOCO and Flickr 30K datasets, but due to large training times that are needed, we decided to persist with theFlickr 8k dataset. In addition, we could experiment with various models and hyperparameters and analyze the results better and in time using the smaller dataset.

2 Related Work

The problem of still image description with natural text has gained interest more recently. Farhadi et al. [1] used detections to infer a triplet of scene elements which is converted to text using templates. Similarly, Li et al. [3] started off with detections and pieced together a final description using phrases containing detected objects and relationships. More powerful language models based on language parsing have been used as well [4, 5]. The above approaches have been able to somehow describe images but they are heavily hand designed and rigid when it comes to text generation.

In this work we combined deep convolutional nets for image classification [6] with recurrent networks for sequence modeling[7].

3 Model

Some of the early approaches to the image caption generator were based on translational algorithms, image encoding and language decoding. The first step is the image feature

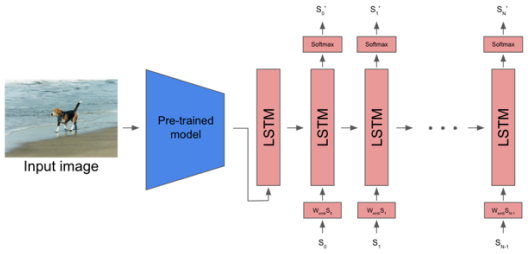


Figure 1. Architecture of Encoder-Decoder model approach.

extraction which mainly requires identifying the objects and its relative location in the image. The second step is to generate a corresponding image description by combining the vectorized image feature with semantic information. In this paper, we followed the same path as that of a neural and probabilistic framework to generate descriptions from images. We split our approach in two parts.

3.1 Image Embedder

For the representation of images, we used a Convolutional Neural Network (Xception, VGG-16). We followed this approach as an initial baseline model using pre-trained CNN models (e.g. Xception, VGG-16) to extract features from images. We used VGG-16 model and Xception model from Keras application from which we slightly modified input and output to make desirable data size. The input image is resized to meet each model's input frame: 224x224 for VGG-16 model and 299x299 for Xception model. The last classification layer from the Xception model is removed to obtain a 2048 feature vector. From the baseline model with default setting, the BLEU score we achieved was merely 15 to 16 percent.

3.2 LSTM based Sentence Generator

The choice of LSTM is governed by its ability to deal with vanishing and exploding gradients [8], the most common challenge in designing and training RNNs. The CNN extracted features were fed to a sequential LSTM model which is responsible for generating image captions.

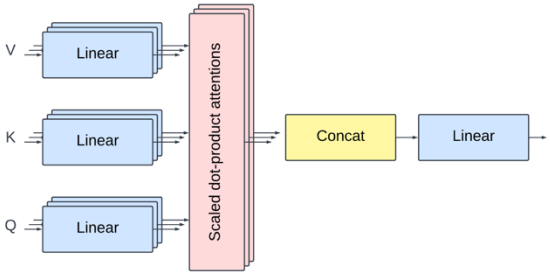


Figure 2: Multihead attention mechanism

The main weakness of this approach is that the important image feature information is dissipated as the time step of the prediction sequence is longer. To overcome this drawback, the attention mechanism that has no longer the sequential process is introduced.

4 Attention mechanism

Attention mechanism replicates human's complex cognitive ability. People can selectively attend to certain parts of an image for quick perception which is a biological mechanism known as human attention. In natural language processing, humans pay attention to keywords when reading long text. The transformer network architecture implements this attention mechanism. Although the transformer architecture is based on an encore-decoder model similar to that of an RNN, the transformer has no time step associated with the input which is the main difference.

Among the multiple different attention mechanisms, we used a multihead attention mechanism. The multihead attention mechanism uses multiple keys, values, and queries to calculate multitude information selected from the input information in parallel. As shown in Fig. 2, the output values are generated from each attention by focusing on different parts of the input information. Then, these output values are concatenated and projected to generate the final value.

5 Experiments

We performed an extensive set of experiments to assess the effectiveness of our model using several metrics and model architectures, in order to compare prior art.

Our model building approach was as follows

1. Create Baseline models using Xception and VGG16 with LSTM for caption generation. Record Corpus BLEU scores for these models
2. Create Baseline models using Inception + Transformer + Attention and various optimizers such as Adam, Adamax, Adagrad and SGD with default hyperparameters.
3. Test models by varying the learning rate and record BLEU scores for each permutation of m models
4. Experiment using various combinations of layers, and input features

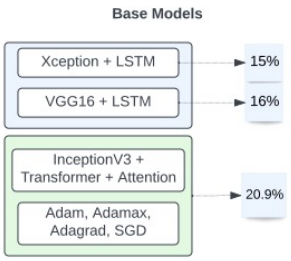


Figure 3. Baseline model performance

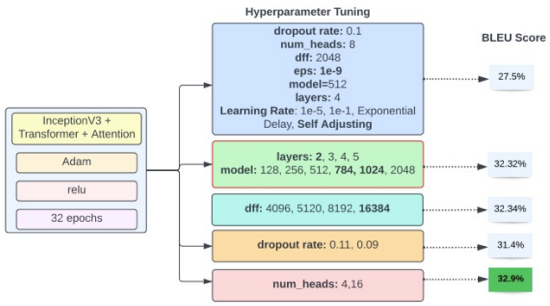


Figure 4. Transformer model hyperparameter tuning process

5. Identify the two best performing models and experiment further using combinations of dff (feed forward upwards project size)
6. Identify the best performing model from above and experiment further using combinations of dropout rate and number of heads for the attention framework.

5.1 Evaluation Metrics

The most commonly used metric so far in the image description literature has been the BLEU score [9], which is a form of precision of word n-grams between generated and reference sentences. The score scale is from 0.0 to 1.0 which represents perfect mismatch and perfect match respectively. From 2 different types of BLEU score, we used corpus BLEU score which calculates the BLEU score for multiple sentences such as a paragraph or a document. The sentence BLEU score, whereas, evaluates a sentence against one or more reference sentences. Besides BLEU, we were exploring the perplexity of the model for a given transcription. The perplexity is the geometric mean of the inverse probability for each predicted word. We did some experiments using this metric to perform choices regarding model selection and hyperparameter tuning using test dataset but we do not report it since BLEU is always preferred.

We also evaluated the models for training loss and accuracies. The expectation was that models with lower training loss and higher accuracies, unless the model is overfitted, will always result in a better fitted model and yield higher BLEU scores. BLEU scores achieved by each model were compared with the Train Loss and Accuracy trends.

5.2 Evaluation Metrics

For evaluation we use a number of datasets which consist of images and sentences in English describing these images. Each image has been annotated by unbiased labels with 5 sentences that are relatively visual.

5.3 Results

The initial trials were with the CNN + LSTM structure. The pretrained CNN models we compared with were VGG16 and Xception. We used default parameter values for the baseline models in which the optimizer was Adam and Relu activation function, the Softmax output function, and Categorical cross entropy loss function were applied. The highest BLEU score we got from the baseline models was 16%. The next round was to test the transformer model with default parameters in which 32 epochs, Relu activation function, 4 layers, 512 model dimension, 2048 feed forward upwards projection size(df), 8 heads, 0.1 dropout rate were applied. With adoption of the transformer model, the BLEU score increased to 21%. Although the BLEU score increased about 5% from the initial baseline model, the score was not high enough. We, therefore, started hyperparameter tuning. The first parameters we adjusted were learning rate and eps and we were able to achieve 27%. The number of layers and the dimension of model were also investigated with fixed learning rate and eps from the previous tuning. With this optimization, we recorded the highest BLEU score of 32%. The dropout rate and the number of heads was also investigated, and the score gained 1% more and ended up as 32.9%.

A complete list of BLEU scores for each model can be found in the appendix A.

Trends of Train Loss and Accuracies across epochs (Figure 3) showed that model with 4 heads for the attention framework, feed forward upwards project size of 16384 and 1024 features generated the least training loss and higher accuracies and expected higher BLEU scores, which is confirmed in the table in Appendix A.

Trends of Train Loss and Accuracies across epochs (Appendix B) showed that model with 4 heads for the attention framework, feed forward upwards project size of 16384 and 1024 features generated the least training loss and higher accuracies and expected higher BLEU scores, which is confirmed in the Table 1.

The models generated some agreeable and some interesting predictions - which can be found under Appendix C in Supplemental material.

References

[1] Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. In: European conference on computer vision, Berlin, pp. 15-29.

[2] Brownlee, J., How to Develop a Deep Learning Photo Caption Generator from Scratch (Jason Brownlee, June 27, 2019), accessed Jan 28, 2022 https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/

[3] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In Conference on Computational Natural Language Learning, 2011.

[4] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In ACL, 2012.

[5] D. Elliott and F. Keller. Image description using visual dependency representations. In EMNLP, 2013.

[6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In arXiv:1502.03167, 2015.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8), 1997.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8), 1997.

[9] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: A method for automatic evaluation of machine translation. In ACL, 2002.

[10] Lindsay, Grace W. “Attention in Psychology, Neuroscience, and Machine Learning.” Edited by Adam H. Marblestone. Frontiers in Computational Neuroscience, no. 2020, 2020. https://doi.org/10.3389/fncom.2020.00029.

A BLEU scores after hyperparameter tuning

Tabel 1. Varying d_model and Layers: 2 layer models with 1024 and 784 features generated the highest BLEU scores when testing on 100 images.

Optimizer	Activation	**Layers**	**D_model**	Dff	Dropout_rate	Num_heads	lr	Eps	MAX BLEU
Adam	relu	2	512	2048	0.1	8	Custom	1E-09	0.319
Adam	relu	3	512	2048	0.1	8	Custom	1E-09	0.306
Adam	relu	4	512	2048	0.1	8	Custom	1E-09	0.293
Adam	relu	5	512	2048	0.1	8	Custom	1E-09	0.248
Adam	relu	2	256	2048	0.1	8	Custom	1E-09	0.315
Adam	relu	3	256	2048	0.1	8	Custom	1E-09	0.293
Adam	relu	4	256	2048	0.1	8	Custom	1E-09	0.295
Adam	relu	5	256	2048	0.1	8	Custom	1E-09	0.293
Adam	relu	2	784	2048	0.1	8	Custom	1E-09	0.322
Adam	relu	3	784	2048	0.1	8	Custom	1E-09	0.245
Adam	relu	4	784	2048	0.1	8	Custom	1E-09	0.198
Adam	relu	2	128	2048	0.1	8	Custom	1E-09	0.314
Adam	relu	3	128	2048	0.1	8	Custom	1E-09	0.307
Adam	relu	4	128	2048	0.1	8	Custom	1E-09	0.274
Adam	relu	2	1024	2048	0.1	8	Custom	1E-09	0.323
Adam	relu	3	1024	2048	0.1	8	Custom	1E-09	0.206
Adam	relu	4	1024	2048	0.1	8	Custom	1E-09	0.156
Adam	relu	2	2048	2048	0.1	8	Custom	1E-09	0.323
Adam	relu	3	2048	2048	0.1	8	Custom	1E-09	0.206

Table 2. Varying dff (feed forward upwards projection size): Models with 1024 features and dff=16384 provided the highest BLEU scores on test data.

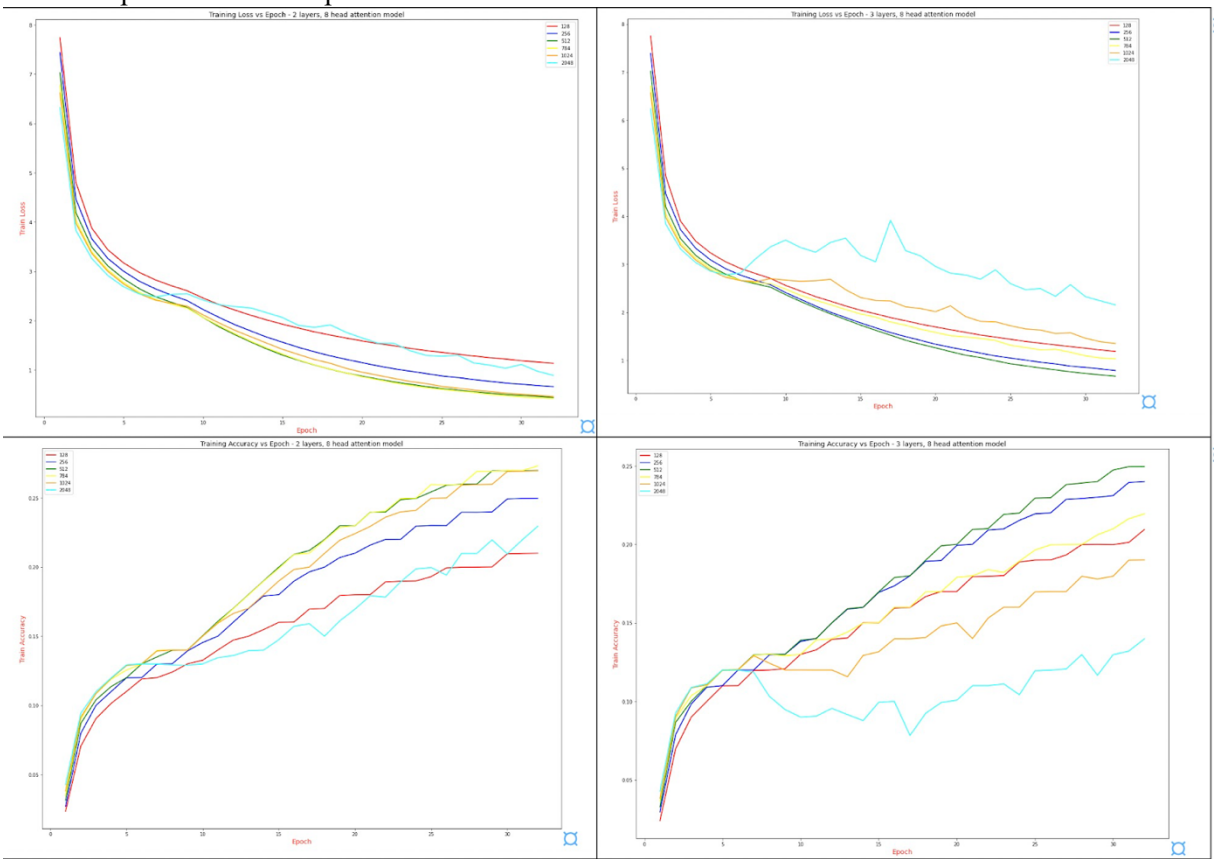
Optimizer	Activation	Layers	D_model	**Dff**	Dropout_rate	Num_heads	lr	Eps	MAX BLEU
Adam	relu	2	784	4096	0.1	8	Custom	1E-09	0.311
Adam	relu	2	1024	4096	0.1	8	Custom	1E-09	0.308
Adam	relu	2	784	5120	0.1	8	Custom	1E-09	0.296
Adam	relu	5	1024	5120	0.1	8	Custom	1E-09	0.308
Adam	relu	3	784	8192	0.1	8	Custom	1E-09	0.299
Adam	relu	4	1024	8192	0.1	8	Custom	1E-09	0.312
Adam	relu	2	784	16384	0.1	8	Custom	1E-09	0.310
Adam	relu	4	1024	16384	0.1	8	Custom	1E-09	0.323

Table 3. Varying dropout rate and number of heads for attention model: It appears that a dropout rate of 0.1 provided the best results. Changing the number of heads for the Attention model to 4 yielded higher BLEU scores

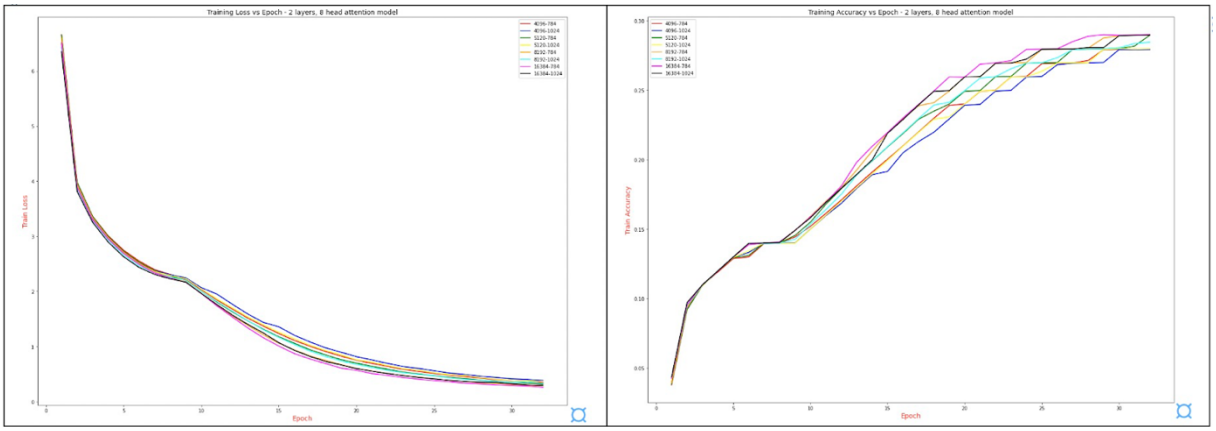
Optimizer	Activation	Layers	D_model	Dff	**Dropout_rate**	**Num_heads**	lr	Eps	MAX BLEU
Adam	relu	2	1024	16384	0.11	8	Custom	1E-09	0.314
Adam	relu	2	1024	16384	0.09	8	Custom	1E-09	0.313
Adam	relu	2	1024	16384	0.1	4	Custom	1E-09	0.329
Adam	relu	5	1024	16384	0.1	2	Custom	1E-09	0.297

B Loss and Accuracy

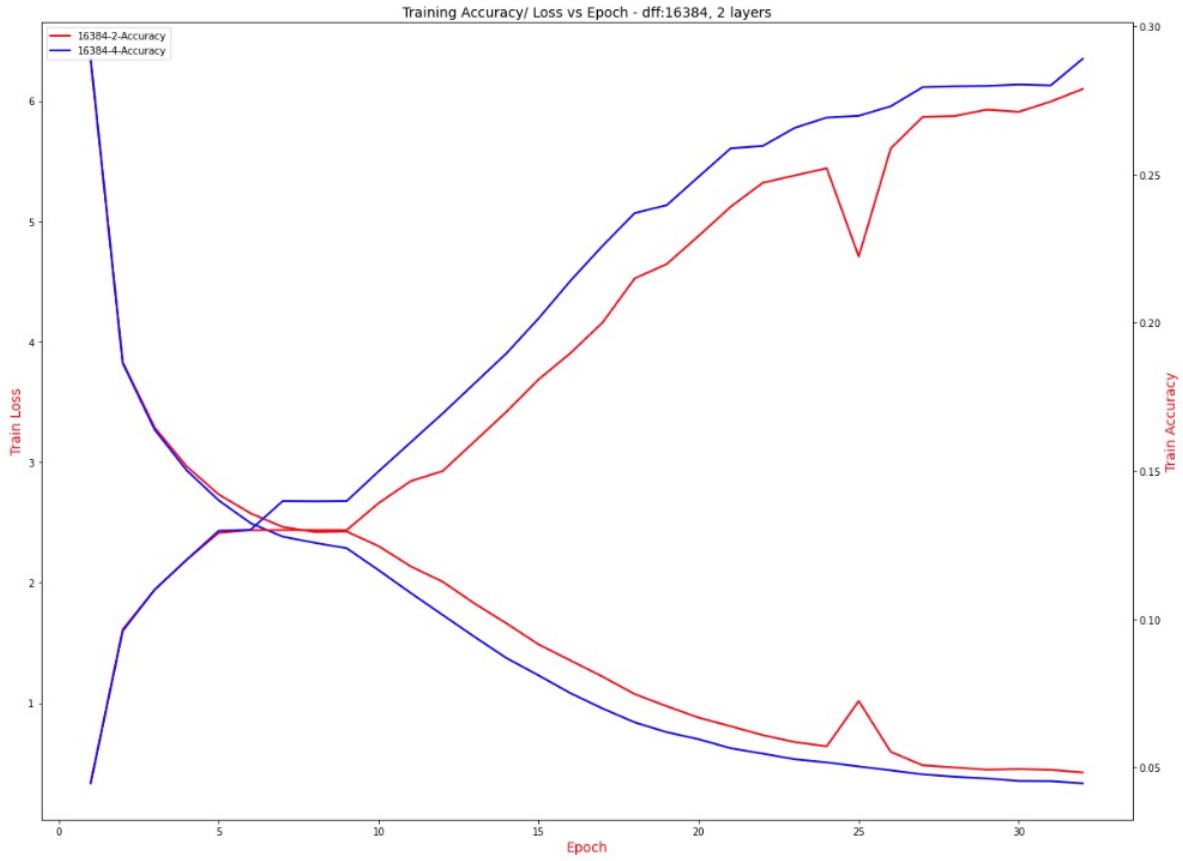
1. 2 and 3 layer models: Training loss is lowest for the model with 784 features in the encoder/decoder inputs. Train accuracy is the highest for the 2 layer model with 784 features. We expect his model to perform the best.



2. 8 head attention model: model with best hyperparameters and dff: 16384, 784 features exhibited the lowest training loss and the highest accuracy. However, highest corpus BLEU scores were observed for the 1024 features model.

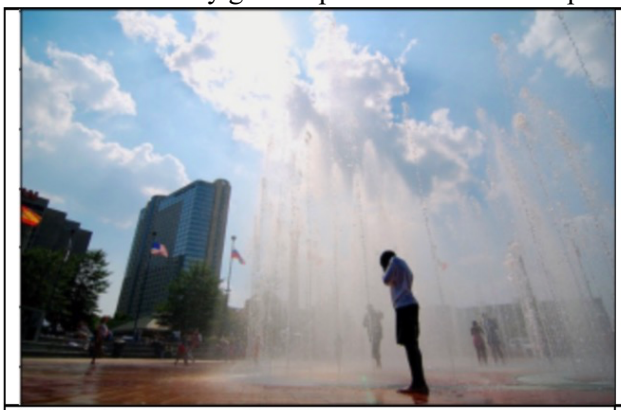


3. Model with 4 heads for the attention framework provided the best results and is also confirmed with the corpus BLEU scores.



C Sample images with predicted captions

- Some really good captions that the model predicted:

	
BLEU-1 score: 0.25 BLEU-2 score: 0.5 BLEU-3 score: 0.6597539553864471 BLEU-4 score: 0.7071067811865476 **Real Caption:** people are playing in water fountains **Predicted Caption:** kids are standing in the sprinklers getting soaked	BLEU-1 score: 0.2857142857142857 BLEU-2 score: 0.5345224838248488 BLEU-3 score: 0.6867198272427282 BLEU-4 score: 0.7311104457090247 **Real Caption:** little girl is petting golden dog **Predicted Caption:** small girl gives kiss to tan dog

- Some not so good ones:

BLEU-1 score: 0.10510841176326924
BLEU-2 score: 0.1966398326430567
BLEU-3 score: 0.2526301062874043
BLEU-4 score: 0.26896050220204015

Real Caption: closeup of white bunny with another white bunny and black horse in the background
Predicted Caption: two white sit on the green grass

BLEU-1 score: 0.27973809117540177
BLEU-2 score: 0.2136534962622895
BLEU-3 score: 0.2976746589772035
BLEU-4 score: 0.3234073084059581

Real Caption: two dogs fight over stick on grassy field lake in the background
Predicted Caption: two dogs are playing in the water

- And some are outright funny!

BLEU-1 score: 0.12383969996431167
BLEU-2 score: 0.3276490485424231
BLEU-3 score: 0.4835356722608987
BLEU-4 score: 0.5329462628216854

Real Caption: two basketball players fighting over control of ball
Predicted Caption: two men are playing in the bathroom

- Expected higher scores for this image

BLEU-1 score: 0.37151909989293497
BLEU-2 score: 0.3276490485424231
BLEU-3 score: 0.2983578836180589
BLEU-4 score: 0.3564026463354183

Real Caption: girl walks on sidewalk while talking on cellphone
Predicted Caption: woman walking while talking on the phone