Multi-Modal Image Captioning

Ruth-Ann Armstrong Thomas Jiang Chris Chankyo Kim
Stanford University
{ruthanna, twjiang, chankyo}@stanford.org

Abstract

Image captioning is a popular benchmark task in the field of computer vision with important real world applications. Our project explores image captioning on Flickr8k, a small image captioning dataset, and applies traditional and state-of-the-art approaches for the task. We experiment with various model architectures including the multi-modal paradigm, pretrained encoders and transformer decoders with multi-headed attention. Our best model, a transformer encoder-decoder network with a pretrained CNN ResNet-18 encoder achieved a BLEU-1 score of 0.879 and a BLEU-4 score of 0.543 on the dataset.

1. Introduction

Over 12 million individuals above the age of 40 in the United States are afflicted with vision impairment. Image captions are invaluable assistive tools that enable these individuals to comfortably navigate their surroundings [1]. With increasingly large populations of society accessing, and in many cases dependent on, the internet, accurate image captioning is a convenient and inexpensive solution to support vision impaired individuals.

In recent years, the topic of image captioning and descriptive imaging has grown in popularity within computer vision. Early methods addressed the issue of captioning in human understandable language with statistical probability language models that utilized handcrafted features or neural networks following an encoder-decoder architecture. Although numerous image captioning techniques and systems exist today, current methods face non-trivial challenges including developing better methods for generating natural structured language akin to human speech and developing clear caption semantics that are consistent with image content. This project aims to improve automated captioning methods by integrating the latest natural language processing (NLP) models alongside traditional visual feature extraction methods to provide enhanced web accessibility.

This project primarily explores variations in both the traditional encoder-decoder architecture and the increasingly popular multi-modal learning paradigm. The motivation to explore a multi-modal architecture stems from the dual-modal nature of the image captioning task. Specifically, captioning incorporates both image processing and speech construction, with the former explored in computer vision while the latter in natural language processing. Multi-modal models have demonstrated aptitude in integrating the joint representations of different modalities, such as vision and language [10], and have potential for improved performance in image captioning.

For both model paradigms, the input and output variables remain constant. Input consists of an individual preprocessed image and its training caption, or caption that represent the ideal RNN input. Training captions are only used during model training and excluded during validation. Output consists of an embedded prediction caption that is translated into its lexical sentence. This predicted sentence represents the predicted caption for the input image and is compared to its ground-truth caption to evaluate model accuracy.

Encoder-decoder and multi-modal architectures differ in their handling of the input image and caption features. Encoder-decoder models apply a convolutional neural network (CNN), such as AlexNet, VGG-16, and ResNet, to extract the input image features. Afterwards, the image features are passed through a decoder network, such as a recurrent neural network (RNN), long-short term memory network (LSTM), or transformers, to construct the predicted caption. In contrast, the multi-modal model explored in this project introduces a multi-modal layer that incorporates image features, token features, and initial embedded tokens. The image features are extracted by passing input images through pre-trained CNN models, akin to encoding in encoder-decoder models. Token features are acquired by passing embedded caption tokens through recurrent network layers.

2. Related Work

We explore different, overlapping categories of models used for image captioning below.

Task Agnostic Multi-Modal Architecture Multi-modal architectures involve mapping image features and text features to the same representation space to produce image captions. One For All (OFA), introduced by Wang et al. [17], is a task agnostic pretrained multi-modal model that achieves state-of-the-art performance on a variety of tasks, including image captioning. This model currently outperforms other methods on the leaderboard for the image captioning benchmark COCO Captions dataset [4] which contains over 300,000 images and over 1.5 million captions (with a CIDER score of 149.6). Transformers [14] are used as the backbone architecture of the model.

To process the input, images are first split into P patches, then projected to features of hidden size H. Text is processed using the byte-pair encoding algorithm, and the resulting subword sequences are then embedded into features. Further, objects in images are represented as a sequence of discrete tokens which capture the label of the object and its bounding box. Images, objects and text are then discretized into a unified output vocabulary and are fed into the transformer to be jointly processed. By doing this, different types of data can be represented in a shared space, which allows the model to be task-agnostic. This feature of the model in addition to its high accuracy makes it particularly exemplary: OFA also achieves competitive accuracies on other multi-modal and uni-modal tasks such as image generation, visual grounding, image classification and language modeling. Improvements made on this model are therefore likely to be beneficial in a variety of domains.

Leveraging Information from Pretrained Language Models Chen et al. [3] make use of linguistic knowledge encoded in large pretrained language models (PLMs) to boost the effectiveness of image captioning models in VisualGPT. The authors propose this type of model for settings where training large models from scratch is infeasible due to limited resources or data. This is relevant for small datasets like Flickr8K. They introduce a self-resurrecting transformer encoder-decoder attention mechanism to allow the model to effectively leverage the linguistic knowledge from pretrained models when making predictions. The authors initialize the parameters of the decoder using a pretrained language transformer model, GPT-2 [12] and randomly initialize the encoder layers.

Another model which leverages knowledge from pretrained language models is a fusion model proposed by Kalimuthu et al. [9]. The model consists of a ConvNet encoder, an LSTM decoder, a BERT [6] encoder and a novel Fusion module which combines representations from the pretrained BERT language model and the image captioning model to produce captions in a way that leverages the knowledge contained in BERT. The authors do this combination by using the hierarchical fusion method described by Fan et al. [7] which involves fusing the hidden states of the two models using concatenations and non-linear activation functions.

Other Transformer Based Architectures Most state-of-the-art models used for image captioning use the transformer architecture as part of their base. Other transformer-based models which have achieved competitive scores on the COCO Captions Benchmark leaderboard include Pan et al. [11] who introduce a unified attention block mechanism and Cornia et al. [5] who exploit connections between high level and low level features using a connectivity structure at the decoder stage to produce a more expressive model.

Reinforcement Learning-Based Techniques Various authors have explored using the reinforcement learning paradigm for generating image captions.

Wei et al. [19] introduce a VisualReserved model which allows for the re-encoding of past visual context in addition to the part of the image that is the current focus of the attention mechanism when generating the text sequence. The authors couple this with an Attention-Fluctuation Supervised model which adds effective matching of visual information to the models reward function using a policy-gradient learning algorithm. This is in contrast to typical attention mechanisms which focus solely on generating attention scores based on the correlation between predicted words and the image.

One disadvantage of using reinforcement learning techniques for caption generation is that generated text tends to be less diverse. Shi et al. [13] combat this by using a partial off-policy learning scheme to encourage the model to explore new possibilities for generated text. To do this, a diverse caption distribution is picked as the exploration behavior policy.

RNNs and LSTMs Prior to the introduction of transformer architectures, multimodal recurrent neural networks (m-RNNs) introduced by Mao et al. [10] were state-of-the-art models for image captioning tasks. At each time step, the model consists of an input word layer, two word embedding layers, a recurrent layer, a multimodal layer and a softmax layer.

In the word embedding layers, inputs are embedded into dense word representations w(t) . Where w(t) is the word representation at time step t, r(t-1) is the recurrent activation at timestep (t-1) and U is a model parameter, the recurrent layer computes:

$$r(t) = \text{ReLU}(U_r \cdot r(t-1) + w(t)) \tag{1}$$

Image representations I are obtained by using computed image features (the authors use FC7 of AlexNet). Outputs

from the word embedding and recurrent layers and the image feature extractor are then combined as follows to produce the final multimodal output:

$$m(t) = g_2(V_w \cdot w(t) + V_r \cdot r(t) + V_i \cdot I) \quad (2)$$

LSTMs, first introduced by Hochreiter et al. [8] are an alternative to RNNs which are better at learning long term interactions in sequence models. Wang et al. [16] were one of the earlier sets of authors to use bidirectional LSTMs for image captioning.

Novel Training Objectives Wang et al. [18] define an explicit Human Consensus-Oriented (objective) to prioritize generating annotations with higher quality. Here, the model is trained with the objective of generating higher rated captions with higher priority. This is a particular interesting approach as it bakes prioritizing the quality of captions into the model, which likely leads to the generation of more descriptive, useful captions for end-user tasks.

3. Dataset

The training, validation, and testing data used for this project is primarily derived from the publicly available Flickr8K collection from Kaggle [2]. This dataset consists of approximately 8,000 images that are each paired with four to five different captions, with approximately 37,000 in total, that clearly describe the salient actors, environment, and events. The images were collected from six different Flickr data groups with a focus on generalized images that avoid well-known people or locations and were manually filtered display various scenes and situations. This dataset was selected due to its popularity in the field of image captioning as a benchmark collection for sentence-based image description. The dataset was separated into a 80/10/10 training/validation/test split (29,600 training, 3,700 validation, and 3,700 test) with batch size of 500 samples.

3.1. Pre-Processing

The base Flickr8K collection provided 8,091 distinct images with varying width and height dimensions and 37,193 sample captions. A separate text file associated the sample captions to their image file path. Due to the constraints of our model architecture, as described below, significant preprocessing of the raw dataset was required.

3.1.1 Image Features

CNN encoding architectures are used in this project to extract features from image data. Due to the structure of CNN input layers, dimensions of the raw input image data is required to remain static. Since the images in the base Flickr8K dataset have varying sizes, pre-processing is required to standardize each image to a fixed shape. In our case, each image was zero padded to match the shape of the maximum width and maximum height found across all images, which was determined to be 500 by 500. The images were also normalized by subtracting the mean pixel value per channel across all images to center and appropriately scale the input data such that each pixel value is in the range [0,1]. Afterwards, the images were resized to a resolution of 224 by 224 to allow compatibility with pre-trained encoder CNN models, such as AlexNet VGG-16 and ResNet-18. Since our group used PyTorch for creating the models, the channel dimensions were switched with the height and width dimensions such that the channel precedes them.

3.1.2 Captions and One-Hot Embedding

RNN and Transformer decoding architectures are used in this project to predict the word tokens for the image captions. Due to the training constraints for training the RNN layers, the caption samples are required to be modeled as a sequence of word tokens, which will eventually be converted into one-hot embedding . Since the captions in the base Flickr8K dataset are provided as single string objects, pre-processing is required to tokenize each caption sentence. In our case, each caption sentence was first filtered for any punctuation or special line characters (i.e. new line) and lowercased. Afterwards, the sentence was tokenized by splitting the words delimited by whitespace characters. Tokenization was completed by adding the start sentence token "<SOS>" and end sentence token "<EOS>" to the list of tokenized words. A deeper inspection reveals the average length of the corpus to be of length 11.3 while the max caption length is 33. For the transformer model, we chose a suitable sequence length of 20 and padded the remaining words with a pad token "<PAD>". This is to ensure our dimensions stayed consistent throughout the transformer decoder network.

3.1.3 Vocabulary and One-Hot Embedding

Tokenized captions were used to create an indexed vocabulary of variable size n consisting of arbitrary unknown token <UNK> and top $n-1$ frequent words throughout all captions. Vocabulary was used to generate one-hot embedded representation of top $n-1$ frequent words in captions. These one-hot embedded word vectors serve as inputs and outputs for the RNN and Transformer decoders.

3.2. Proposed Data Augmentation

Although data augmentation was not employed at the time of writing this report, our group proposes a cross-synonym substitution method of expanding the quantity of sample captions available for training. Natural language

words often have numerous synonymous words that can be substituted for the original word while retaining the integrity of the original sentence. Using this idea, our group proposes the construction of a cross-synonym dictionary that maps each tokens in the vocabulary to their synonyms found among the $n-1$ vocabulary tokens. For example, if the word tokens `jump` and `leap` are included in the vocabulary, then the cross-synonym dictionary will map `jump` to `leap` and `leap` to `jump`. These cross-synonym associations can then be used to replace words in original captions to create additional augmented variations. As an example, suppose a sample caption is

```
<SOS> Kids jump over the fence <EOS>
```

Following the method described above, an additional augmented caption can be created

```
<SOS> Kids leap over the fence <EOS>
```

and added to the training corpus.

4. Models and Methods

In the following sections, we describe the baseline, encoder-decoder, and multi-modal image captioning architectures evaluated in this project. The following described models were batch trained for 50 epochs under a 80/10/10 train/validation/test split (except for Transformer models) with the Adam optimizer, learning rate $\alpha = 0.05$, batch size $n = 500$ samples, and cross entropy loss criterion.

4.1. Baseline Model

Our baseline model follows an encoder-decoder architecture that uses a convolutional neural network encoder to extract image features, which are passed in as input to the recurrent neural network decoder array. The encoder model is a three layer convolutional neural network of output channels 12, 12, and 24 and kernel size 3, two max pooling layers of kernel size 2, dropout training layer of probability 0.2, and a final fully connected layer and logarithmic softmax output. The decoder model consists of an l-layered recurrent neural network with sequential long short-term memory layers, where l is the size of the longest caption. The encoder and decoder models are connected by the feature output by the fully connected layer in the CNN and the first LSTM layer in the RNN module, where the CNN output serves as the LSTM initial hidden state (See Figure 1).

4.2. Pre-Trained Encoder Networks

We introduce an additional variation of the baseline encoder-decoder model that utilizes a pre-trained convolutional neural network encoder. The aim of this modification is twofold. First, the additional models serve as additional baselines to compare results. Second, the pre-trained encoder serves as a representation of generalized image feature extraction. As such, comparing the preliminary baseline and other models with this model will provide insight to how our models will perform relatively outside of the provided dataset. For instance, the pre-trained VGG-16 convolutional network was one of the chosen encoders to replace our original CNN module in the baseline model. The VGG-16 network was chosen for its well-known high performance accuracy (92.7%) in the ImageNet dataset, a collection consisting of over 14 million images and 1000 classes. To accomodate the modification, the dataset preprocessing step was also modified to resize the images to a strict 224 by 224 dimension to be compatible with the VGG-16 input layer.

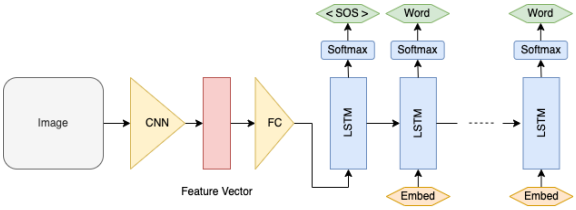


Figure 1. Encoder-Decoder/Baseline model architecture. Image features are extracted by CNN encoder into a single feature vector passed into long short-term memory layers in RNN decoder

4.3. Transformer Encoder-Decoder

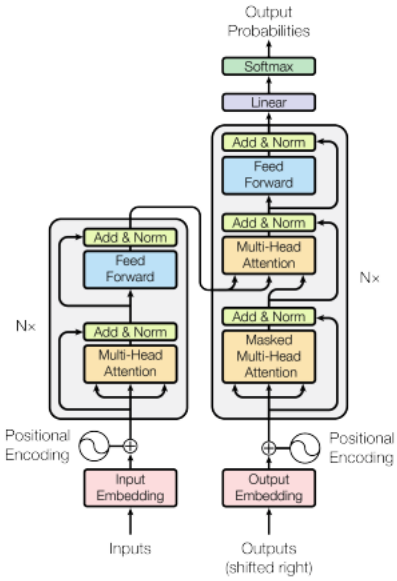


Figure 2. Transformer encoder-decoder architecture from Attention is all you need [15]: For our experiments, input embedding are the pretrained CNN on different layers.

Model Name	Encoder layer	Decoder layer	n heads	lr	Caption length
T1	4	8	16	$1e^{-4}$	20
T2	4	4	8	$1e^{-4}$	20
T3	2	4	8	$1e^{-4}$	20
T4	2	2	4	$1e^{-4}$	20
T5	2	1	2	$1e^{-4}$	20

Table 1. Transformer Encoder-Decoder experiments with pre-trained CNN ResNet-18 Encoder and Transformer Decoder

In this model, image features are extracted using the pre-trained ResNet18 model from as the CNN encoder. ResNet18 was chosen for its exemplary performance and popularity in image classification with relatively small datasets. Since ResNet18 is commonly used within image classification tasks, the final layers of the model include a softmax activation layer. In order to extract the raw image features directly from the convolution layers, the last two layers of the pre-trained ResNet18 model were removed.

Similar to the pre-trained encoder-decoder network design, this model primarily substitutes transformer modules as the decoder instead of basic RNN and LSTM cells. The internal architecture of the transformer encoder-decoder model used in this project is a variation of the model described in Figure 2.

Positional Encoding was used to highlight relative token order to the model and cross-entropy loss was used for training purposes. Since transformers are relatively more complex decoders, the train/valid/test was split in 85/10/5 to ensure the model did not overfit on small data. The transformer model was trained for 50 epochs using dropout rate of 0.2 to add even more regularization. This is because in pre-experiments, our team noticed a large gap between training and validation loss which suggests overfitting to data. We hypothesize this is due to the small dataset and depth of the network; this will be explained in more detail in the results/evaluation and conclusion section. In particular we choose 5 different experiments which have a better performance in pre-experiments. These 5 experiments are highlighted in Table 1.

$$P_{ij} = \begin{cases} \sin(i \cdot 10000^{-\frac{j}{d}}) & \text{if j is even} \\ \cos(i \cdot 10000^{-\frac{j-1}{d}}) & \text{otherwise} \end{cases}$$

.

4.4. Multi-modal Architecture

The multi-modal architecture used in this project introduces a new multi-modal layer that accepts a concatenated input vector consisting of image features, word token features, and word token embeddings. The image features are extracted from pre-trained CNN encoder modules similar to ones used in the encoder-decoder architectures described above. The word token features are extracted by passing the input word token through an initial embedding layer and a recurrent network layer. In our experiments, the recurrent network layer consisted of variations in the basic RNN and LSTM cells and Transformer modules. The word token embedding is simply the embedding vector retrieved after passing the input word token through the initial embedding layer and before it is passed into the recurrent network.

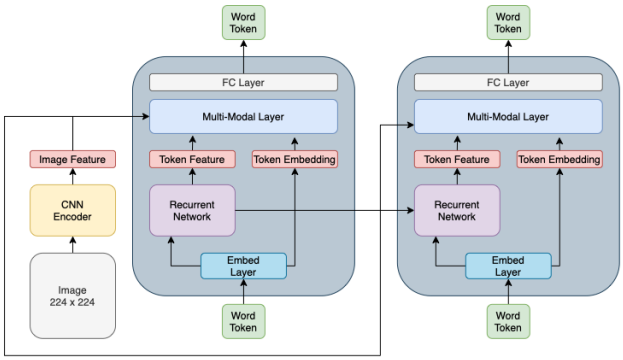


Figure 3. Multi-Modal Architecture inspired by Junhua et. al. [10]. CNN Encoder layer is varied between different pre-trained models. Recurrent Network layer is interchanged between simple RNN and LSTM to Transformer models. Multi-modal cell is denoted by navy box

These three features are concatenated in accordance to the multi-modal output function described in Equation 2 from the Related Works section. The function g_2 used by Junhua et. al. [10] consisted of a $\tanh$ activation as a non-linear concatenation. The multi-modal model introduced in this project uses a rectified linear unit (ReLU) activation function in precaution of vanishing gradients. The resulting output of the multi-modal layer is passed through a final fully-connected linear layer and its softmax activation to produce the predicted word token. The above process denotes the completion of one pass over the multi-modal cell, from which the hidden weights of the recurrent network layer is passed onto the next recurrent network layer of the following multi-modal cell. This sequence is followed until the entire prediction caption is complete.

5. Results/ Evaluation

5.1. Quantitative Results

The primary metric used to evaluate predicted caption accuracy is the BiLingual Evaluation Understudy (BLEU) scores of 1 and 4 n-grams with their referenced target captions. Comparing BLEU scores, it is immediately evident that the baseline encoder-decoder and VGG16-RNN models have significantly worse performance compared to other models. To understand this result, it is important to note that

Model	BLEU-1	BLEU-4
Baseline Encoder-Decoder*	0.222	0.000
VGG16-RNN*	0.194	0.000
AlexNet-RNN	0.609	0.013
ResNet18-RNN	0.612	0.013
VGG16-LSTM	0.617	0.014
AlexNet-LSTM	0.615	0.013
ResNet18-LSTM	0.621	0.012
ResNet18-Transformer(T1)	0.847	0.510
ResNet18-Transformer(T2)	0.860	0.537
ResNet18-Transformer(T3)	**0.879**	**0.543**
ResNet18-Transformer(T4)	0.835	0.502
ResNet18-Transformer(T5)	0.791	0.387
MM-VGG16-RNN	0.612	0.162
MM-VGG16-LSTM	0.617	0.171
MM-ResNet18-LSTM	0.621	0.165

Table 2. Average BLEU-1 and BLEU-4 scores of different model architectures. Baseline Encoder-Decoder and VGG16-RNN highlighted with $*$ to indicate that subset of full dataset used. Transformers trained for 50 epochs.

for the baseline encoder-decoder and VGG16-RNN models, a reduced training dataset, with only up to 500 images and maximum caption length of 8) was used in attempts to accelerate model development and iteration, which may have contributed to subpar performance. The baseline model achieved a BLEU-1 of 0.222 and BLEU-4 of 0.0, equivalent to correctly predicting 2 word tokens in its 8 word caption and not finding a single correct sequence of words in the caption. Similarly, the VGG16-RNN model achieved a BLEU-1 of 0.194 and equivalent BLEU-4 of 0.0, slightly under performing compared to the baseline model. Observing the training and validation loss history (see Appendix), the baseline model demonstrates a tighter generalization gap compared to that of VGG16-RNN, indicating that the VGG16-RNN is overfitting to the training data.

In comparing the BLEU-1 and BLEU-4 scores of the other various encoder-decoder architectures, it is evident that there is minor differences between exchanging different CNN encoder modules. In particular, the difference between the BLEU-1 scores of AlexNet-RNN and ResNet18-RNN was about 0.003 and the difference between BLEU-4 scores is 0. Similarly, the difference between the BLEU-1 scores of VGG16-LSTM, AlexNet-LSTM and ResNet18-LSTM is at most 0.006 and the difference between BLEU-4 scores is at most 0.002. This indicates that there is no measurable difference between using different pre-trained encoder modules to extract image features. Similarly, no significant difference in performance was found between RNN and LSTM decoder modules. Specifically, the difference between BLEU-1 scores for AlexNet-RNN and AlexNet-LSTM was only 0.006 and the difference between ResNet18-RNN and ResNet18-LSTM was only 0.009.

The transformer encoder-decoder models performed the best not only out of the encoder-decoder architectures but all models explored in this project. In particular, the hyperparameter-optimized ResNet18-Transformer T3 proved to be the best model with BLEU-1 of 0.879 and BLEU-4 of 0.543, significantly higher than any other model performance. Furthermore, the BLEU-4 score of 0.543 is particularly promising, as it indicates that this transformer model is successful in predicting, on average, half of the correct word token sequences across the tested captions.

Interestingly, the multi-modal models exhibited mediocre performance, residing between the RNN/LSTM encoder-decoder models and transformer encoder-decoder models. The BLEU-1 scores of MM-VGG16-RNN, MM-VGG16-LSTM, and MM-ResNet18-LSTM are fairly similar to those of the RNN/LSTM encoder-decoder models. However, the BLEU-4 scores of the multi-modal models are approximately a whole magnitude larger than those of the RNN/LSTM encoder-decoder models. Despite this improvement, the multi-modal models perform slightly worse than the transformer encoder-decoder architectures. A plausible explanation is that the multi-modal design itself demonstrates superior architecture to the simple encoder-decoder models but performs worse than the transformer encoder-decoder models because it is still using the basic RNN/LSTM cells for its recurrent network layers. For future work, it would be interesting to pursue a multi-modal transformer model and evaluate its performance to the transformer encoder-decoder model.

5.2. Qualitative Analysis of Captions

This section observes 6 select samples of generated captions using the best performing model, ResNet18-Transformer (T3), in order of increasing BLEU-4 scores.

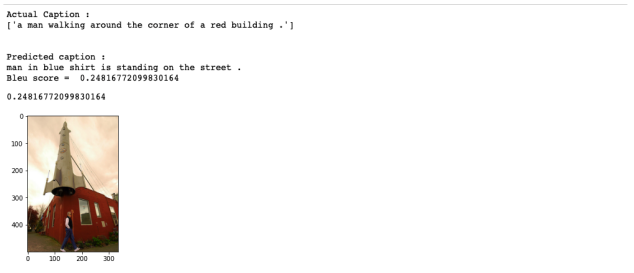


Figure 4. Sample image with BLEU-4 captioning score 0.248

The prediction model performed the worst on the image from Figure 4 with a BLEU-4 score of 0.248. The predicted caption ignores the red building in the image and incorrectly describes the color of the shirt of the person in the

picture. Additionally, subject's action is described incorrectly as standing rather than walking, which is likely because a snapshot of a person walking is visually similar to that of a person standing. One hypothesis for the incorrect guess about the color of the man's shirt is a fault in the attention mechanism: the color blue accurately describes the color of his pants, so it is possible that the mechanism attended to the wrong portion of the image when generating the color descriptor for his shirt. A possible reason that the building was ignored in the caption, is because the similarity between its hue and the background of the image might have resulted in the model interpreting it as a non-salient portion of the image.

Figure 5. Sample image with BLEU-4 captioning score 0.264

Next, the prediction model captioned the image from Figure 5 with a BLEU-4 score of 0.264. Here, the model incorrectly describes the picture as having two dogs present, and does not include any description about the chain in the image. Additionally, the caption generated is one that is repeated for other images with similar color schemes in the dataset. This might mean that the CNN portion of the model was not effective at recognizing more fine-grained differences between visually similar pictures, which led to incorrect generalizations about the correct descriptions of pictures with these similarities.

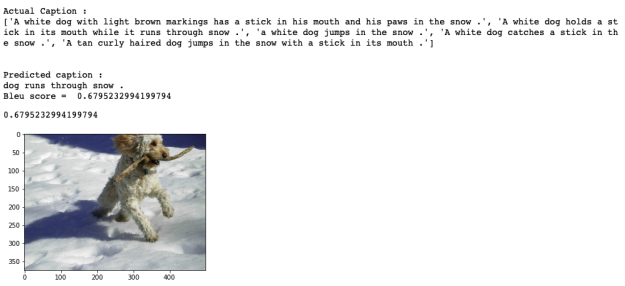


Figure 6. Sample image with BLEU-4 captioning score 0.680

Next, the prediction model captioned the image from Figure 6 with a BLEU-4 score of 0.680. Though the generated caption for this image is accurate, it is less descriptive than the gold standard captions. This lack of descriptiveness might be remedied with more effective finetuning of our CNN model so that more fine-grained features of the images are captured.

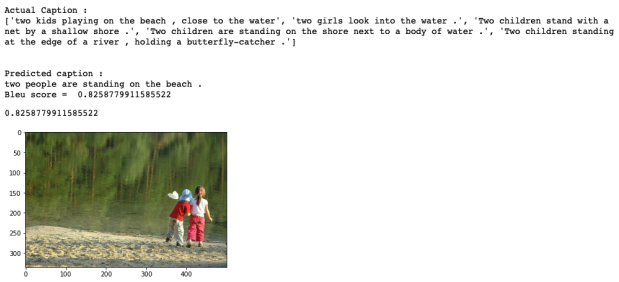


Figure 7. Sample image with BLEU-4 captioning score 0.826

Next, the prediction model captioned the image from Figure 7 with a BLEU-4 score of 0.826. The caption generated by the model for this image achieves a relatively high BLEU-4 score. Hypotheses for why the model might have done well at predicting an accurate caption is that there is a high degree of contrast between the subjects of the image and the background, and that there is not much ambiguity in what they are doing in the picture.

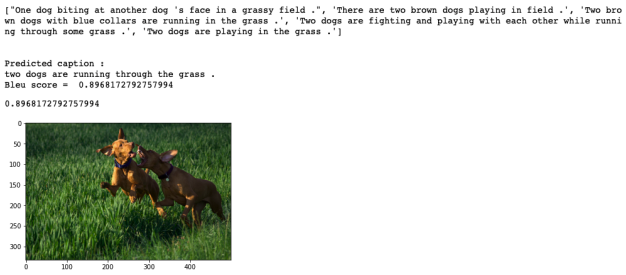


Figure 8. Sample image with BLEU-4 captioning score 0.897

Next, the prediction model captioned the image from Figure 7 with a BLEU-4 score of 0.897. This example illustrates an instance of a less descriptive (though accurate) repeated caption being used to describe an image. This is likely also because of the problem described for *Figure 5*.

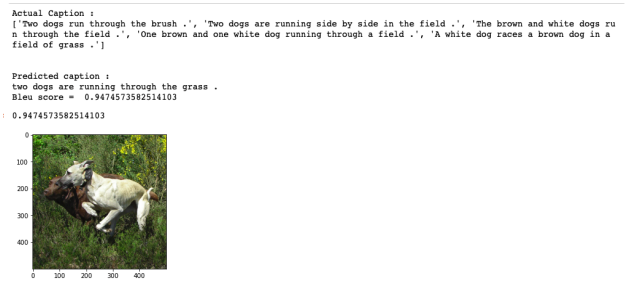


Figure 9. Sample image with BLEU-4 captioning score 0.947

Next, the prediction model captioned the image from Figure 7 with a BLEU-4 score of 0.947. Though this is an

example of a repeated caption, the model achieves a high BLEU-4 score for this image. Interestingly, the caption here has the same characteristic lack of descriptiveness as the other examples described, but achieves a high accuracy because it happens to be similar to one of the gold standard labels. This shows that the accuracies of models on image captioning benchmarks can be highly sensitive to the quality and diversity of the captions in the training and testing datasets.

5.3. Error Analysis

5.3.1 Repeated words and phrases

During caption generation, we noticed words were repeated many times. Similar pictures often had similar words. We suspect that the attention mechanism was paying attention to similar areas multiple times. For deeper inspection and for future work, we could introduce an attention mechanism loss together with our model which would help our attention mechanism pay attention to different parts of the image.

5.3.2 Attention to wrong object

Moreover, we noticed that our attention mechanism often paid attention to wrong spots of the image. Larger pre-trained CNNs such as ResNet-152 would probably extract higher detailed features from images which could potentially solve the problem. Fine tuning our current CNN even further may also prove to be fruitful. For future work, we could experiment with different pre-trained CNN's or apply transfer learning such that pre-trained CNN's are able to learn on specific datasets too. Deeper networks, however, require more data to reduce overfitting - thus training on larger datasets such as Flickr30k or COCO is helpful.

5.3.3 Different images had similar captions

Finally we also noticed that different images with similar color schemes had similar captions. This is evidenced above "dogs are running through the grass" which was accurate for two of the three images. This suggests that our CNN is able to be furthermore improved via fine-tuning or upgrading to a deeper pretrained CNN such as ResNet-152.

6. Conclusion

Overall, we found that while we were able to achieve relatively effective accuracies under the BLEU-1 (0.879) metric with a majority of our model architectures, the models did not perform as effectively under the BLEU-4 (0.543) metric. One contributor to this is likely the relatively small size of our training dataset. Our best performing model (T3) which had less parameters than T1 and T2 achieved higher BLEU-1 and BLEU-4 scores with a smaller dataset. A hypothesis for moving forwards is that if we would want to experiment similar architectures on larger datasets, deeper networks would be more performant. The RNN/LSTM multi-modal architectures we explored demonstrated improved performance over the traditional RNN/LSTM encoder-decoder paradigm but performed worse than the transformed encoder-decoder models, indicating superior architectural design but limited by worse performance of RNN/LSTM relative to transformers.

A future direction that we are interested in exploring is how to better tailor existing models to contexts where there is a lower amount of data available for training. Transformer architectures outperformed simpler architectures like those which used RNNs or LSTMS as decoders to produce image captions. This is likely as a result of the higher degree of expressivity of transformer models, as well as their self-attention and multi-headed attention mechanisms which are able to focus on specific parts of the input images when generating captions. Exploring larger datasets such as Flickr30k and COCO may also prove to be worthy of time and effort as well as larger CNN's.

Another future direction that our group is interested in is to use transformers as the primary recurrent layer in our multi-modal architecture to compare with the encoder-decoder results demonstrated in this project. Superior performance with multi-modal transformer models would strongly conclude that multi-modal architecture is superior to encoder-decoder models.

References

[1] Fast facts of common eye disorders, 2020. 1

[2] Flickr 8k dataset, 2020. 3

[3] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning, 2021. 2

[4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015. 2

[5] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 2

[7] Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. *CoRR*, abs/1805.04833, 2018. 2

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. 3

[9] Marimuthu Kalimuthu, Aditya Mogadala, Marius Mosbach, and Dietrich Klakow. Fusion models for improved image

captioning. In *International Conference on Pattern Recognition*, pages 381–395. Springer, 2021. 2

[10] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Explain images with multimodal recurrent neural networks. *CoRR*, abs/1410.1090, 2014. 1, 2, 5

[11] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. *CoRR*, abs/2003.14080, 2020. 2

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 2

[13] Jiahe Shi, Yali Li, and Shengjin Wang. Partial off-policy learning: Balance accuracy and diversity for human-oriented image captioning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2167–2176, 2021. 2

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 2

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 4

[16] Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. Image captioning with deep bidirectional lstms. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 988–997, New York, NY, USA, 2016. Association for Computing Machinery. 3

[17] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework, 2022. 2

[18] Ziwei Wang, Zi Huang, and Yadan Luo. Human consensus-oriented image captioning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 659–665, 2021. 3

[19] Yiwei Wei, Chunlei Wu, ZhiYang Jia, XuFei Hu, Shuang Guo, and Haitao Shi. Past is important: Improved image captioning by looking back in time. *Signal Processing: Image Communication*, 94:116183, 2021. 2

7. Contributions & Acknowledgements

1. Chris Kim: Multi-modal architecture, skeleton code, RNN and LSTM architecture, data-preprocessing, write-up

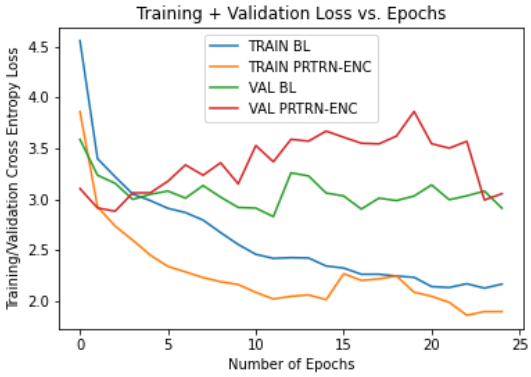
2. Thomas Wang: Transformer architecture, skeleton code, hyperparameter search, generation of qualitative data, write-up

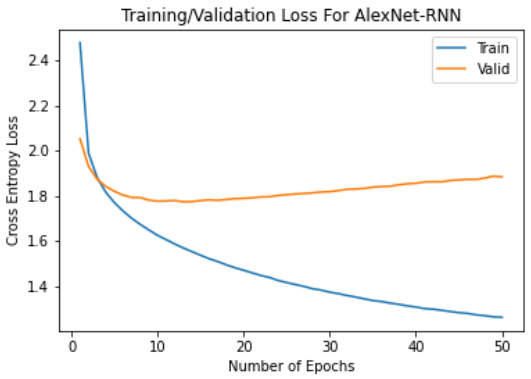
3. Ruth-Ann Armstrong: Initial setup on Github, literature review, write-up

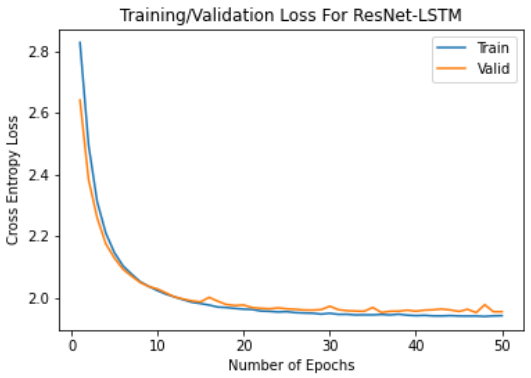
8. Appendix

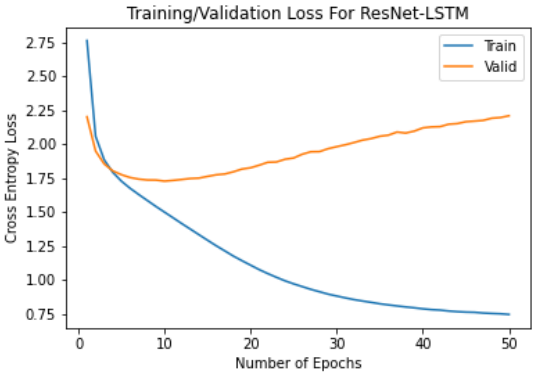
8.1. Training/Validation Loss

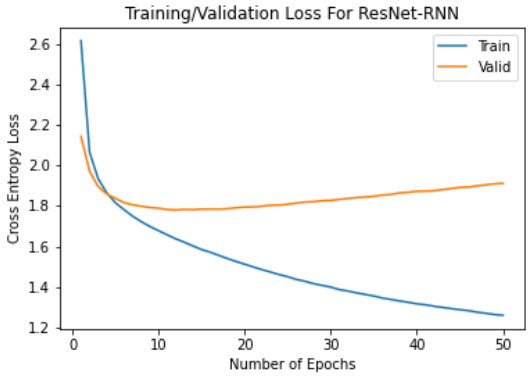
The following graphs are sample training/validation loss values acquired throughout the process of this project. Not all model training/validation loss graphs are presented due to errors with cloud computing services.

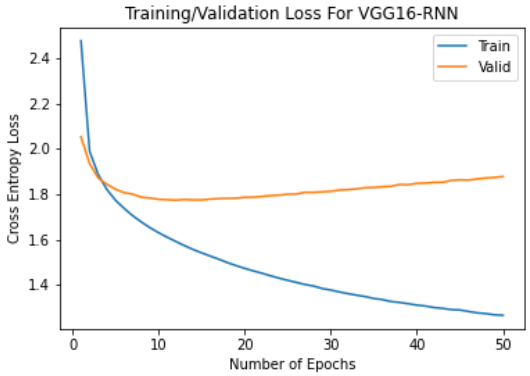


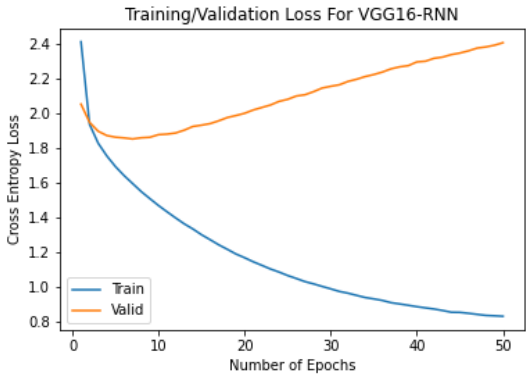





Training/Validation Loss For ResNet-LSTM
Train
Valid
Cross Entropy Loss
Number of Epochs


Training/Validation Loss For ResNet-RNN
Train
Valid
Cross Entropy Loss
Number of Epochs


Training/Validation Loss For VGG16-RNN
Train
Valid
Cross Entropy Loss
Number of Epochs


Training/Validation Loss For VGG16-RNN
Train
Valid
Cross Entropy Loss
Number of Epochs