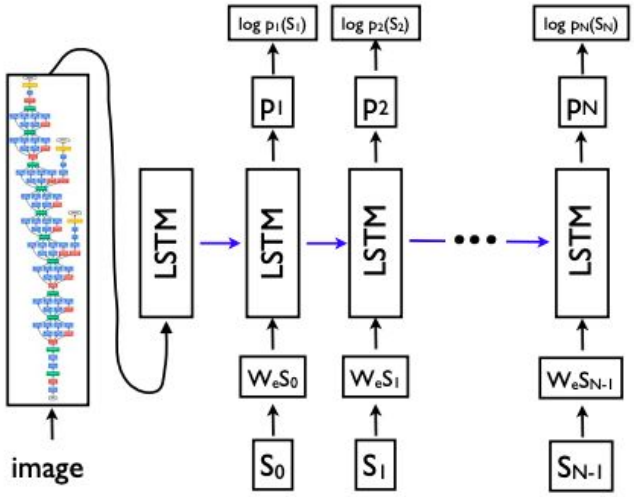
Hugh Yeh
Taiga Hasegawa
Jordan Lovett

CS398 Final Project
Show and Tell
Github: https://github.com/TaigaHasegawa/Deep-Learning/tree/master/caption-generation

Introduction and Previous Studies:
The fundamental goal of computer vision is to analyze and interpret images and the fundamental goal of natural language processing is to understand and generate coherent syntax that demonstrate fluency in an language. To address these fundamental goals, previous papers published, *Show and Tell* (Vinyals et. al.) and *Show, Attend, and Tell* (Xu et. al.), have attempted to connect computer vision to natural language processing. This correspondence is established through recognition of different objects in an image and their relationship to each other through usage of an convolutional neural network encoder, word embeddings and LSTM decoder. In *Show, Attend, and Tell*, an extra "Attention" network is implemented to prioritize objects in an image.



Vinyals et. al.

Encoder:
The encoder used is a deep convolutional neural network pretrained for image recognition. The encoder is capable of extracting features for further processing by LSTMs (and Attention). In *Show, Attend, and Tell* VGGNet pretrained on ImageNet for image recognition as the encoder. For our project, we tests multiple different encoders pretrained on ImageNet using a fine tuning schema in which the entire encoder network is trained throughout the duration of the training

cycle. We found that although slightly slower, this approach nearly guarantees model improvement, whereas other fine tuning strategies proved to be overly sensitive. The encoders used were ResNet 101 and ResNet 152.

Attention:
An Attention module was introduced by *Show, Attend and Tell* and this method allowed for better prioritization and selection of specific objects in an image that are necessary to describe and caption an image. Simply put, Attention is a fully connected network that applies weights and a bias terms to an encoded term, based on the decoder's previous hidden state. The Attention module utilizes previous object detections and word generations to direct and give priority to that object when formulating the next part of captions, therefore, focusing on the relationship between the main object and surrounding objects. This is done by calculating the probability of which aspect of the image should be focused on for the generation of the next word in the caption, and using this probability distribution to determine the object and area of the image that has the greatest likelihood of being important for the generation of future words. This method is similar to the process of writing a sentence where a subject is first defined and then followed by an action. The Attention module is implemented after the encoder and at every LSTM to direct the formulation of the caption.

Decoder:
Just like the specific convolutional networks we used as decoders are able to deal with vanishing and exploding gradients, we choose to use an LSTM in order to deal with the same challenges that occur while training RNNs. An LSTM is an RNN which consists of two states (cell state and hidden state), and 3 gates which determine those states in lieu of new inputs. Its been empirically shown that because LSTMs usually have Each state can be thought of as representing long or short term memory, with the hidden state (the output state implemented in most RNNs) representing short term memory, and the cell state (unique to LSTM variants) representing long term memory. The cell state pushes past information through the network, without a great deal of alteration due to the fact that there are two gates that regulate which information gets concatenated to the cell state. This allows information from one temporal state to get passed along to further states without too much distortion, thereby facilitating the concurrent evaluation of information from even distant temporal states. This feature makes LSTMs extremely useful for textual analysis in which things like pronouns, prepositions and adjectives are given at the beginning of a text, and hence are implied for the remainder of the passage. For image captioning however, the extremely long term memory of the LSTM is only one of its greatest assets, since the majority of captions do not exceed a dozen words in length (although it still clearly would yield improvement over traditional RNNs). As emphasized by Vinyals et al, LSTMs allow us to deal with degenerating gradients, which can both slow or entirely inhibit training of the model.

Evaluation Method:
In this project, we attempt to revisit papers with various network architectures pretrained on ImageNet and pre-established embeddings. In our paper, we compare different combinations of pre-trained and pre-established architecture and their performance using Encoder-Decoder models described by *Show and Tell* and *Show, Attend, and Tell*.

In our specific case, we used LSTMs with both pretrained word embeddings, and embeddings that we trained from scratch. Although in the paper published by Vinyals, et al,the claim is made that pretrained word embeddings would fail to yield better results, we found that performance was improved by nearly a full percentage point while using Glove embeddings. We also used teacher forcing during the training cycle, which we discontinued during the evaluation phase. Training and testing was conducted using the Flickr30k dataset where train and test sets were distributed according to Karpathy, A., et. al. Flickr30k split.

One of the more difficult points of the implementation was the fact that we only used features from the input image once. This meant that the features needed to have the same dimension as the word embeddings. For this, we modified the final layers of the pretrained ResNet, also changing the standard max pooling to an adaptive average pooling.

To evaluate the model, we used BLEU:1 and BLEU:4. In BLEU:N, we compute the percentage of i-gram (where i = 1:N) that occured both the hypothesis and reference. This is a measurement of precision. While this method is a standard used in natural language processing, it is important to note that it has weaknesses such as the fact that the "quality" of the reference is subjective to opinion.

Results from *Show and Tell* and *Show, Attend, and Tell*:
Combinations of Pretrained Modules:

Model	Description
Model 1	Encoder-Decoder Model: *Show and Tell* Encoder: ResNet 152
Model 2	Encoder-Decoder Model: *Show, Attend, and Tell* Encoder: ResNet 101 (no fine-tuning)
Model 3	Encoder-Decoder Model: *Show, Attend, and Tell* Encoder: ResNet 152
Model 4	Encoder-Decoder Model: *Show, Attend, and Tell* Encoder: ResNet 152 Decoder: GloVe Embeddings

Results and Discussion:

Model	BLEU1	BLEU4
Model 1	0.6083560	0.1943750
Model 2	0.6253283	0.2128531
Model 3	0.6365740	0.2284490
Model 4	**0.6431489**	**0.2338500**

Our Sample Images and Captions:

Figure 1.0
Model 1: “a dog is jumping over a stone wall”
Model 2: “a dog is jumping over a stone wall”
Model 3: “a person is walking down a stone walkway”
Model 4: “two people are holding a rope in front of a stone building”
Our Interpretation: “Taiga’s dog standing next to a stone statue”

Figure 2.0
Model 1: “a man is sitting at a table in a restaurant”
Model 2: “a man is sitting at a table with a drink”
Model 3: “a young man in a striped shirt”
Model 4: “a man in a black and white striped shirt is cutting something”
Our interpretation: “Taiga drinking from a glass”

From these sample pictures and from the results the BLEU assessment, we show the strengths and weaknesses of *Show and Tell* and *Show, Attend, and Tell.* When using the model for just

Show and Tell, we see that the model attempts to generalize the different objects in the images into a cohesive idea. By taking into consideration the different objects in the background and objects not associated with the main focus of the image, the model attempts to find a general idea to explain how these different objects are related such as deciding that the environment was a restaurant in Figure 2.0. This model begins to fail as it may make too many assumptions about how the objects are related and over generalize. Furthermore, it may also miss the main action of the image, such as in Figure 2.0 where the main focus is Taiga drinking from a glass. The benefits of *Show, Attend, and Tell* is primarily due to the ability of the model to prioritize objects that are directly being used or near the main object or main focus. As shown in Figure 2.0, this model realizes the main focus is Taiga, and as a result, prioritizes objects being used by Taiga, such as his shirt and utensils near his hand. This allows for greater depth in terms of detail of the main focus of the image, such as describing Taiga's shirt, but this fails when the model attempts to be too specific and doubtful of the object/action such as the model overstating Taiga's action with his hand in Figure 2.0 and the action of the rope in Figure 1.0. Finally, it is also important to note that using pretrained embeddings also improved both BLEU1 and BLEU4 scores. This may be due to the fact that having pretrained embeddings from a large aids with a developing more fluent flow of syntax. This is under the assumption that the words used for pretraining is, to some degree, reflective and representative of the training set, and as we see in some examples as shown above, when this degree of representation is not high enough, pretrained embeddings may actually hurt performance for certain images.

Future Work:

Since we found that pretrained embeddings actually improve performance, we would be curious to see what kinds of captioning effects are produced by pretraining embeddings on various types of textual sources. Could we manipulate the syntactic structure of our captions through using a variety of pretrained embeddings? We would also like to spend more time optimizing the fine-tuning process of pretrained convolutional networks. We would like to compare the results of 1. Training only the last layers of the network 2) Gradually training deeper layers in our network as gradients of final layers begin to converge 3) Training the entire network from end to end. Since we have found the most success in using ResNet152 with pretrained embeddings with full end-to-end fine-tuning, we could use our results from this configuration as a benchmark.

References:

Karpathy, A., et al. *Deep Visual-Semantic Alignments for Generating Image Descriptions.* 2015. arXiv:1412.2306

Vinyals, et al. *Show and Tell: A Neural Caption Generator.* 2015. arXiv:1411.4555

Xu, Kelvin. *Show, Attend and Tell: Neural Image Caption Generator with Visual Attention*. 2015. arXiv: 1502.03044