Sequenced Show, Attend, and Tell: Natural Language from Natural Images

Marcus Comiter
mcomiter@g.harvard.edu

Surat Teerapittayanon
steerapi@seas.harvard.edu

June 15, 2016

1 Abstract

We present Sequenced Show, Attend, and Tell: Natural Language from Natural Images, a machine translation-inspired framework to perform automatic captioning from images. Given an input image of a scene, our algorithm outputs a fully formed sentence explaining the contents and actions of the scene. Our method uses an LSTM-based sequence-to-sequence algorithm with global attention for generating the captions. The input to our algorithm is a set of convolution features extracted from the lower layers of a convolutional neural network, each corresponding to a particular portion of the input image. Follow this, using a global attention model, the inputs are used to generate the caption one word at a time with the LSTM "focusing" on a portion of the image as dictated by the attention model.

We compare our proposed method with a number of different methods, including the attention-based method of Xu et al. (2015) as well as the attention-less method of Vinyals et al. (2015). Additionally, we present results both with and without the use of pretrained word embeddings, with the use of different CNNs for feature extraction, the use of reverse ordering of the source input into the LSTM, and the use of residual connections. We find that our proposed method is comparable with the state of the art. Further, we find that the use of pretrained word embeddings, different CNNs, reversing the ordering of the input, and the use of residual connections do not have a large impact on system performance.

2 Introduction

As the number of images continues to grow, the need to be able to summarize these images in an efficient and scalable manner grows in tandem. One such method for doing so is through the automatic generation of captions. Automatic generation of captions encompasses many difficult sub-problems, including object recognition, activity recognition, and the generation of coherent text approximating that generated by humans.

In this paper, we introduce Sequenced Show, Attend, and Tell, a machine translation-based method for automatic caption generation from images. The core of our proposed method is a sequence-to-sequence based LSTM neural network that uses the concept of *global attention* to focus attention on different portions of the image in generating captions one word at a time.

Our methodology differs from previous methods applied to this task in two main ways. First, our model differs from methods that use a single feature that represents the entire image globally in that we use as input multiple features extracted from the input image, all of which are used in a given combination at each time step as dictated by a global attention model, which weights each of the features derived from lower levels of the CNN at each time step. Second, our method differs from other attention-based models through its use of global attention and an input-feeding approach as described in Luong et al. (2015). These differences, and the full models themselves, are presented in detail in Section 4.

We present results on the Flickr8k dataset (Rashtchian et al. (2010)), and generate features on the Flickr30k and Microsoft COCO datasets. Beyond presenting results comparing our method to other caption generation methods on these datasets, we extend our analysis to study how results are impacted by the use of pretrained word embeddings, the use of different CNNs for feature extraction, the use of reverse ordering of the source input into the LSTM, and the use of residual connections.

In Section 3, we describe the problem formally. In Section 4, we describe our proposed model. In Section 5, we present experimental results. We discuss the results and findings in Section 6. We describe future work in Section 7. Finally, we conclude in Section 8.

3 Problem Description

We now formally describe the problem we seek to solve. Given a set of images $\mathcal{I}_{\text{training}}$ and corresponding captions $\mathcal{C}_{\text{training}}$, where each caption is of variable length n and made of up a sequence of words $w_1, \ldots, w_n$ where each w_i is drawn from a vocabulary $\mathcal{V}$, we seek a model that can correctly generate a caption given an unseen image.

For all models presented and compared with in this paper, the features from which the caption is generated is not the image itself, but rather a feature set $\mathcal{F}_{training}$ extracted using a Convolutional Neural Network (CNN). For some models, this feature set consists solely of a single feature extracted from a fully connected layer of the CNN. For other models, the feature set consists of multiple features extracted from a lower convolutional layer of the CNN.

Given these features and corresponding captions, we train our models. Once the models have been trained, we use them to generate captions for the test portion of the dataset. We evaluate the models using BLEU score presented in Papineni et al. (2002) (specifically presenting BLEU-1, BLEU-2, BLEU-3, and BLEU-4, where n in BLEU-n corresponds to the the fact that we use all x-grams of size $x = 1$ through $x = n$ in calculating the BLEU score). In the following section, we present detailed summaries of our proposed model as well as the two models against which we compare our proposed model.

4 Model and Algorithms

4.1 Proposed Method: Sequenced Show, Attend, and Tell

4.1.1 Feature Generation using Convolutional Neural Networks

In our proposed method, the feature set $\mathcal{F}$ used as input to the LSTM (which is described in the subsection below) is generated from a lower convolutional layer of a Convolutional Neural

Network (CNN). To do this, the input image I is used as input to a pretrained CNN, and the feature set is derived from the resulting value at a selected convolutional layer. Extracting features in this manner has the effect of generating multiple features, rather than a single feature, for a given image, where each feature corresponds to a particular convolution window on the image. Figure 1 shows for an example CNN (in this example, VGG-16) an example of the layer from which our proposed model extracts features. In this particular example, the features are extracted from the final convolution layer before the fully connected layer.

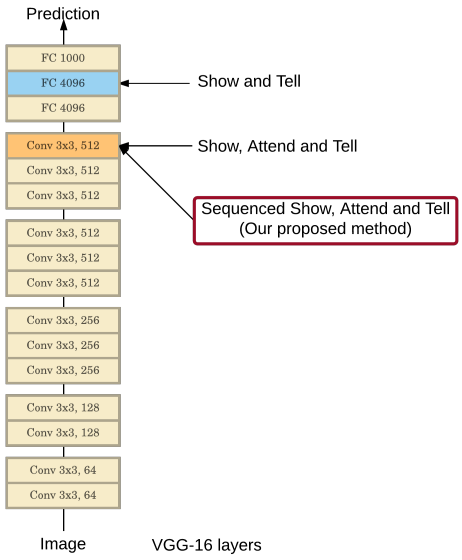

Figure 1: Layers of the CNN from which features are generated.

4.1.2 Sequence-to-Sequence Model

The core of our proposed Sequenced Show, Attend, and Tell method is a machine-translation based sequence-to-sequence model with global attention, as introduced in Luong et al. (2015). Formally, under this framework, our model seeks to find the conditional probability $p(C|F)$, where C corresponds to the generated caption made up of words $w_1, \ldots, w_n$, and F corresponds to the features $f_1, \ldots, f_n$ extracted from the CNN during the preprocessing step described in the subsection above. The model has both an encoder and a decoder. The encoder finds a representation $\mathbf{s}$ for the source input (the features from the CNN). The decoder generates the caption word by word.

In our implementation of the sequence-to-sequence model, we use a Stacking Long Short-Term Memory (LSTM) neural network, where the training objective is:

$$J_t = \sum_{(C,F) \in D} -\log p(C, F)$$

where D is the set of training data.

4.1.3 Global Attention

Our proposed method uses *Global Attention* in deciding on which portions of the image to focus in generating the caption word by word. Under this model, the hidden states of all of the source positions are potentially used in generating a word at a given timestep.

We now formally define this global attention method used in our model. At each time step, the LSTM has a hidden layer $\mathbf{h}_t$. Under the global attention model, we wish to find a context vector $\mathbf{c}_t$ that takes into account the sources at all positions in the multi-step source that is used as input to the LSTM. This context vector $\mathbf{c}_t$ is then combined with the hidden layer $\mathbf{h}_t$ to produce an attention hidden state:

$$\widetilde{\mathbf{h}_t} = \tanh(\mathbf{W}_c[\mathbf{c}_t; \mathbf{h}_t])$$

Under the global attention model, the context vector $\mathbf{c}_t$ is derived through the following process. First, an alignment vector $\mathbf{a}_t$ is calculated by comparing the current time-step's hidden state $\mathbf{h}_t$ with each of the source hidden states $\bar{\mathbf{h}_s}$ such that

$$\mathbf{a}_t = \frac{\exp(\mathbf{h}_t^\top \mathbf{W}_a \bar{\mathbf{h}_s})}{\sum_{s'} \exp(\mathbf{h}_t^\top \mathbf{W}_a \bar{\mathbf{h}_{s'}})}$$

Once $\mathbf{a}_t$ has been calculated, the context vector $\mathbf{c}_t$ is calculated as a weighted average of the alignments:

$$\mathbf{c}_t = \mathbf{a}_t^T \mathbf{h}_s$$

Additionally, our method adopts the "Input-feeding Approach" described in Luong et al. (2015). This approach seeks to rectify the shortcoming of the model in that a coverage set is not explicitly maintained to account for which portions of the source have already been addressed (i.e., for which portions of the image a word in the caption has been generated). The input-feeding approach addresses this problem by concatenating the attention-added hidden states $\widetilde{\mathbf{h}_t}$ with the inputs of the next time steps. This allows the model to be cognizant of previous alignment choices.

4.1.4 Caption Generation

The global attention model described in the previous subsection is used to calculate a context vector for each time step of the caption generation. In order to utilize the context vector, at each time step, the context vector is combined with the current target state $\mathbf{h}_t$ to produce a distribution over the vocabulary for the target word w_t such that:

$$p(w_t | w_{<t}, x) = \text{softmax}(\mathbf{W}_s \widetilde{\mathbf{h}_t})$$

where $w_{<t}$ represents the previously generated words before timestep t (i.e., conditioning on the previously generated output) and $\mathbf{W}_s$ are learned weights. This allows for the generation of the caption, the output of our proposed Sequenced Show, Attend, and Tell model. In generating the caption, we use Beam Search.

4.2 Show and Tell Method

As discussed further in Section 5, we compare our proposed method against two other methods: the first, "Show and Tell: A Neural Image Caption Generator" (Vinyals et al. (2015)) is discussed

in this section, and the second, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention" (Xu et al. (2015)) is discussed in the following section.

The "Show and Tell: A Neural Image Caption Generator" presented in Vinyals et al. (2015) serves as a baseline method against which we compare our method, as this model is simpler as it does not involve any notion of attention.

Like our proposed method, the Show and Tell model similarly uses a CNN to generate the input for the model. However, in contrast to both our proposed method and the method of Xu et al. (2015) presented in the following section, rather than generating multiple features from a given image, the input image is instead encoded as a single vector of fixed length using a CNN. This single feature can be extraced, for example, from one of the fully connected layers in the CNN. This is shown in Figure 1 for the VGG-16 CNN, where the feature is extracted from the penultimate fully connected layer in the CNN.

Importantly, as there is only a single feature generated by the CNN, there is no concept of attention used in this model. Instead, an LSTM neural network is trained, into which this single feature is then fed.

We now take a moment to compare this model to our proposed model and that of Xu et al. (2015). This model differs from the other two models mainly in the respect that it considers only a single feature representation of the image and does not include the notion of *attention* that is present in both our proposed Sequenced Show, Attend, and Tell model and the Show, Attend, and Tell Method of Xu et al. (2015) described in the next section.

4.3 Show, Attend, Tell Method

The most direct comparison to our proposed method is the algorithm presented in Xu et al. (2015). Unlike the model of Vinyals et al. (2015), in which a single encoding vector is extracted as a feature, the Show, Attend, and Tell method extracts vectors at lower levels in the CNN, such that an encoded vector corresponds to a particular portion of the input image (this is the same process that we follow in our proposed method).

With these encoding vectors calculated, the vectors are fed in order to an LSTM using two different types of attention models: hard and soft attention. Under a hard attention model, a single convolution feature is the focus of a given timestep. In a soft attention model, a number of convolutional features are used at each timestep, where the relative weighting among the candidate features corresponds to the probability that the area is the "correct" place on which attention should be focused.

The attention model is used in a similar manner to our proposed method. Upon choosing an attention model f_{att}, the method derives a weighting vector α for the inputs using a multi-layer perceptron conditioned on the hidden state of the previous timestep h_{t-1} via the following:

$$e_{ti} = f_{att}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

The context vector for timestep t $\hat{\mathbf{z}}_t$ is then calculated as

$$\hat{\mathbf{z}}_t = \phi(\{\mathbf{a}_i\}, \{\boldsymbol{\alpha}_i\})$$

where ϕ is the function that returns a single vector given the input features $\mathbf{a}$ and the weighting vector $\boldsymbol{\alpha}$.

Given the context vector, the LSTM is used to generate hidden states that are then used to generate a probability distribution over the vocabulary. The model used is the following:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + i_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

where $\mathbf{i}_t$ is the input state, $\mathbf{f}_t$ is the forget state, $\mathbf{c}_t$ is the memory state, $\mathbf{o}_t$ is the output state, $\mathbf{h}_t$ is the hidden state, $\hat{\mathbf{z}}_t$ is the context vector described previously, $\mathbf{E}$ is an embedding matrix, σ is the logistic sigmoid activation function.

The LSTM is run using the extracted features as the source inputs. This process forms the caption one word at a time, using the attention model to focus on a particular portion of the image at each timestep.

We now describe how the Show, Attend, and Tell model relates to the previously described models. Importantly, this model differs from both our proposed method as well as the previously discussed Show and Tell method of Vinyals et al. (2015). This model differs from our proposed method in that our method uses the "Input-feeding Approach" described earlier that gives additional leeway in constraints of how the different parts of the image is used, while this method uses a "doubly attentional" approach that requires the model to give equal attention to all parts of the input image (e.g., all features). This model differs from the method of Vinyals et al. (2015) in that it uses attention and multiple features from a single input image.

5 Experiments

In this section, we present extensive experimental results on the Flickr-8K dataset of Rashtchian et al. (2010). Additionally, we present results showing how modifications to our proposed method, including the use of pretrained word embeddings, the choice of feature generating CNN, reversed source input, and residual connections impact performance. Where possible, we provide direct comparison to the methods presented in Vinyals et al. (2015) and Xu et al. (2015).

For all experiments, we use the following parameters: weights are initialized uniformly between -0.1 and 0.1, the learning rate is 1, the maximum norm of the gradient vector is capped to be 5, the dropout probability is 0.3, the learning rate is decayed by 0.5 when either validation set perplexity does not decrease or the epoch is beyond 9, the source vocabulary size is 512, the target vocabulary size is 8388 for Flickr-8K, there are 500 hidden states in the LSTM and 2 layers in the encoder/decoder, and the word embedding dimension is 50. We use 5 beams in the Beam Search procedure. All training occurs using a GPU. More specifically, we use a GeForce GTX Titan X 12GB Graphics Card GPU. For the Flickr-8K experiments, we use a batch size of 100 with 20 epochs. Each of the following experiments we present in this section took approximately 3-4 hours to train. Finally, we adapt the seq2seq-att implementation of Kim (2016).

For each set of results, we use BLEU score as a metric (Papineni et al. (2002)). Specifically, we present BLEU-1, BLEU-2, BLEU-3, and BLEU-4, where n in BLEU-n corresponds to the the fact that we use all x-grams of size 1 through n in calculating the BLEU score. To calculate the BLEU scores, we use the script available at the link in this footnote[1].

Our data, including preprocessed convolutional features for the Flickr-8K, Flickr-30K, and Microsoft COCO datasets, is available at the link in this footnote[2]. In forming the training, validation, and test portions of the datasets, we use the splits suggested in Karpathy and Fei-Fei (2015). Our github repository can be found in our project site at the link in this footnote [3],[4]. Finally, we use the pretrained model of VGGNet of Saito (2015) , and the pretrained model of ResNet of Kudo (2016).

Examples of image/caption pairs can be found in our project page, for which a link was provided earlier. Two examples of words and the associated visualisation of the areas in which the attention model focuses in generating the word at its given timestep can be seen in Figure 2.

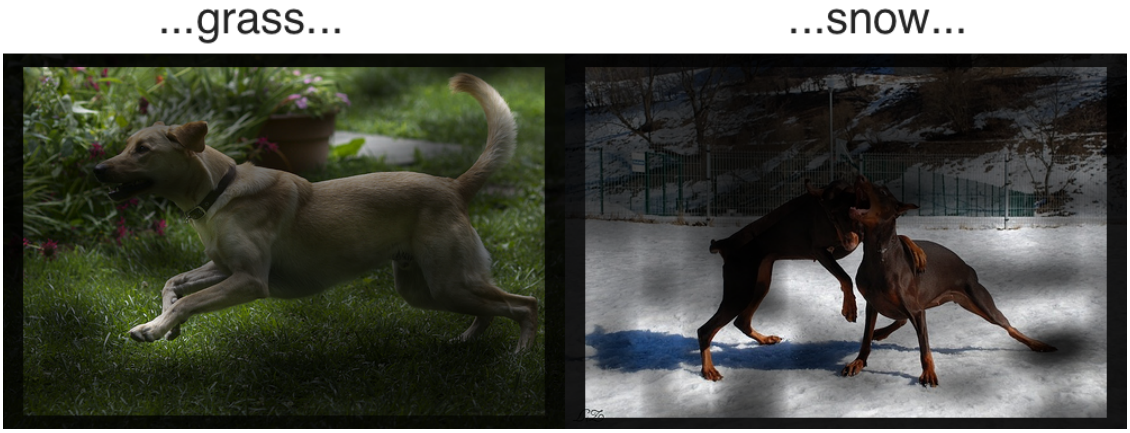

Figure 2: Example words from captions generated with our proposed Sequenced Show, Attend, and Tell method, where each word of the caption is presented above a visualization of the areas in which the attention model focuses in generating the word at its given timestep. Note that in both examples, the attention focuses on the grassy field when generating the word "grass" in the caption "A brown dog is running through the grass" and the snowy background when generating the word "snow" in the caption "Two brown dogs play in the snow."

We first discuss the performance of the three models on the Flickr-8K dataset using VGG-16 and no pretrained word embeddings. This dataset consists of 8,000 images, of which 6,000 are used in the training portion of the dataset, 1,000 are used in the validation portion of the dataset, and 1,000 are used in the test portion of the dataset. For each image, the dataset contains 5 reference caption sentences associated with the image. For training purposes, we break the five reference captions into five separate training examples, such that we have a total of 30,000 data points in the test set (but only 6,000 unique images). For generating results in terms of BLEU-score, the five reference captions are used as a single data point.

[1] https://github.com/karpathy/neuraltalk/blob/master/eval/multi-bleu.perl
[2] https://drive.google.com/folderview?id=0Byyuc5LmNmJPQmJzVE5GOEJOdzQ&usp=sharing
[3] `http://steerapi.github.io/seq2seq-show-att-tell/`
[4] `https://github.com/steerapi/seq2seq-show-att-tell`

Table 1: Results comparing our proposed method with the Show, Tell Method Vinyals et al. (2015) and Show, Attend, Tell Method Xu et al. (2015)

		BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**
Algorithm	Show, Tell	52.5	34.0	21.4	13.7
	Show, Attend, Tell	67.0	44.8	29.9	19.5
	Our Proposed Method	60.8	42.4	29.0	19.8

The results of our method as compared with the methods of Vinyals et al. (2015) and Xu et al. (2015) are shown in Table 1. For all results in this table, features are generated using VGGnet (Simonyan and Zisserman (2014)). For the model of Vinyals et al. (2015), we use the data available from Karpathy and Fei-Fei (2015)[5]. This data includes the encodings of the images calcuated by the CNN, as well as reference captions. The features are generated from the penultimate fully connected layer, and are of dimension 1x4096. For both our proposed model and the model of Xu et al. (2015), we generate the features ourselves by extracting them from the last convolution layer of the CNN. The features are of size 14x14x512, which we flatten to form 196 features of length 512.

As these results show, our proposed method outperforms the method of Vinyals et al. (2015), achieving BLEU scores that are on average 10 points higher than that of the Show and Tell method. When compared with the method of Xu et al. (2015), our method is comparable in all metrics except for BLEU-1. Further, the perplexity of our model for the training and validation portions of the dataset are is shown in Figure 3. We see that the perplexity on the training set after 20 epochs is 13.31, and the perplexity on the validation set after 20 epochs is 26.12. Furhter, the average sentence length for our model is 9.32, and the average sentence length of the reference is 10.86.

We now demonstrate how attention is used in generating the captions for our proposed Sequenced Show, Attend, and Tell method. Figure 2 demonstrates how attention is used in generating the caption. We see in the first image that the attention focuses on the grassy field when generating the word "grass" in the caption "A brown dog is running through the grass," and in the second image that the atention focuses on the snowy background when generating the word "snow" in the caption "Two brown dogs play in the snow."

We now discuss a series of additional experiments, demonstrating the impact of using pretrained word embeddings, different CNNs for feature generation, reversing the ordering of the source vectors, and the use of residual connections. Except where noted, all experiments use VGG-16 and pretrained word embeddings.

We first discuss the impact of the use of pretrained word embeddings. We use the pretrained word embeddings of Pennington et al. (2014). The result of this embedding is that the words are embedded into a 50-dimensional space. We generate results for both keeping the pretrained word embeddings fixed, as well as letting them change during training. The results are shown in Table 2. As these results show, the use of pretrained word embeddings (both fixed and variable) have only a small impact on model performance.

We further experiment with using different CNNs for generating features. We compare the model performance using features generated with VGGNet and ResNet-152. We note that ResNet-152 is larger than VGGNet. The results for each of these different networks is shown in Table 3. As these results show, the use of VGGNet gives a small performance boost for BLEU-1 score, and

[5]The features can be obtained at http://cs.stanford.edu/people/karpathy/deepimagesent/

Table 2: Results comparing the use of fixed, variable, and no pretrained word embeddings (results using features generated from VGGNet)

		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Embedding Type	No Pretrained Embeddings	60.8	42.4	29.0	19.8
	Variable Pretrained Embeddings	61.0	42.0	28.0	18.6
	Fixed Pretrained Embeddings	60.4	42.4	28.5	19.0

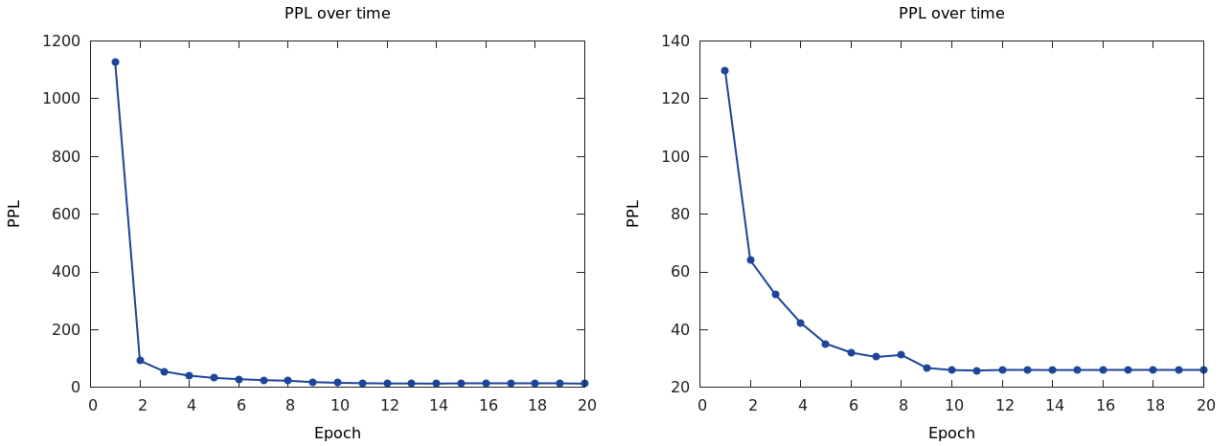

Figure 3: Perplexity over epochs for our proposed model for training (left) and validation (right) portions of the dataset.

comparable results for BLEU-2, 3, and 4.

We now examine the impact of ordering on the model performance. We reverse the order of the source sequence as input to the LSTM. The results are shown in Table 4 comparing the original ordered vs reversed ordered sequence. As these results show, reverse ordering does not improve model performance.

Finally, we examine the impact of using residual connections, in which the hidden states of the the layers are added together and input to the LSTM layer. The results are shown in Table 5. As these results show, the use of the residual connections does not markedly impact performance.

6 Discussion

We were surprised to find that many of the additional experiments we ran had little impact on overall system performance.

Table 3: Results comparing our proposed method with using different CNNs for feature generation (VGG vs. ResNet-152). All results are with using pretrained word embeddings.

		BLEU-1	BLEU-2	BLEU-3	BLEU-4
CNN	VGG-16	61.0	42.0	28.0	18.6
	ResNet-152	60.2	41.8	28.3	18.8

Table 4: Results comparing the effect of ordering (all results with VGG and pretrained word embeddings)

		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Ordering	Normal	60.8	42.4	29.0	19.8
	Reverse	60.7	42.6	29.3	19.9

Table 5: Results comparing the effect of using a residual connection (all results with variable pretrained word embeddings)

		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Residual	No Residual Connections	60.8	42.4	29.0	19.8
	Residual Connections	60.0	41.4	27.7	18.3

Additionally, we were surprised by the long training time of using the Flickr-30K and Microsoft COCO datasets. While we generated features on both of these datasets with the intention of presenting results on these datasets in addition to Flickr-8K, we found that the training this model on these datasets took too long (it was taking days without completing), even using a GPU.

7 Future Work

As discussed previously, we unfortunately did not have enough time to train the model on the Flickr-30K and and Microsoft COCO datasets, as we had been training the models for days without the training process finishing. While we did generate features for these datasets (where the URL for these features is given in Section 5), we found that the model took too long to generate results. Given more time and computational resources, we would generate results for these datasets and see how the extra amount of training data improved model performance.

8 Conclusion

In this paper, we have proposed Sequenced Show, Attend, and Tell, a new methodology for automatically generating captions from images. Under our proposed methodology, a global attention model is used in conjunction with anLSTM to generate a caption word-by-word from a given image. The features used as input to the model are extracted from a convolutional layer of a CNN.

We present results on the Flickr-8K dataset, comparing our method to other recent competitive models. We find that our proposed methodology outperforms the method of Vinyals et al. (2015) and is competitive with the state-of-the-art method of Xu et al. (2015). Further, we find that the use of pretrained word embeddings, different CNNs for generating features, reversing the source input, and the use of residual connections do not markedly impact model performance.

References

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*, pages 3128–3137.

Kim, Y. e. a. (2016). Sequence-to-sequence model with lstm encoder/decoders and attention. https://github.com/harvardnlp/seq2seq-attn.

Kudo, Y. (2016). Deep residual network implementation by chainer. https://github.com/yasunorikudo/chainer-ResNet/.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. (2010). Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147. Association for Computational Linguistics.

Saito, S. (2015). chainer-imagenet-vgg. https://github.com/mitmul/chainer-imagenet-vgg.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

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*, pages 3156–3164.

Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.