Show and Tell: A Neural Image Caption Generator

Oriol Vinyals
Google
vinyals@google.com

Alexander Toshev
Google
toshev@google.com

Samy Bengio
Google
bengio@google.com

Dumitru Erhan
Google
dumitru@google.com

Abstract

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. In this paper, we present a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation and that can be used to generate natural sentences describing an image. The model is trained to maximize the likelihood of the target description sentence given the training image. Experiments on several datasets show the accuracy of the model and the fluency of the language it learns solely from image descriptions. Our model is often quite accurate, which we verify both qualitatively and quantitatively. For instance, while the current state-of-the-art BLEU-1 score (the higher the better) on the Pascal dataset is 25, our approach yields 59, to be compared to human performance around 69. We also show BLEU-1 score improvements on Flickr30k, from 56 to 66, and on SBU, from 19 to 28. Lastly, on the newly released COCO dataset, we achieve a BLEU-4 of 27.7, which is the current state-of-the-art.

1. Introduction

Being able to automatically describe the content of an image using properly formed English sentences is a very challenging task, but it could have great impact, for instance by helping visually impaired people better understand the content of images on the web. This task is significantly harder, for example, than the well-studied image classification or object recognition tasks, which have been a main focus in the computer vision community [27]. Indeed, a description must capture not only the objects contained in an image, but it also must express how these objects relate to each other as well as their attributes and the activities they are involved in. Moreover, the above semantic knowledge has to be expressed in a natural language like English, which means that a language model is needed in addition to visual understanding.

Most previous attempts have proposed to stitch together existing solutions of the above sub-problems, in order to go from an image to its description [6, 16]. In contrast, we would like to present in this work a single joint model that takes an image I as input, and is trained to maximize the likelihood $p(S|I)$ of producing a target sequence of words $S = \{S_1, S_2, \ldots\}$ where each word S_t comes from a given dictionary, that describes the image adequately.

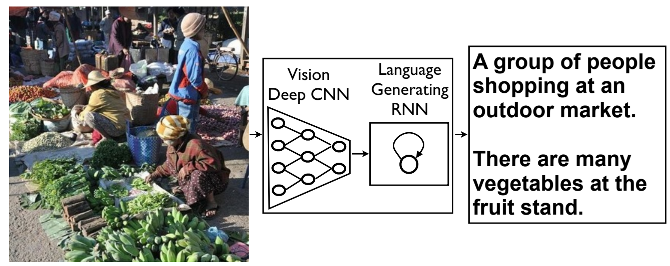

Figure 1. NIC, our model, is based end-to-end on a neural network consisting of a vision CNN followed by a language generating RNN. It generates complete sentences in natural language from an input image, as shown on the example above.

The main inspiration of our work comes from recent advances in machine translation, where the task is to transform a sentence S written in a source language, into its translation T in the target language, by maximizing $p(T|S)$. For many years, machine translation was also achieved by a series of separate tasks (translating words individually, aligning words, reordering, etc), but recent work has shown that translation can be done in a much simpler way using Recurrent Neural Networks (RNNs) [3, 2, 30] and still reach state-of-the-art performance. An "encoder" RNN *reads* the source sentence and transforms it into a rich fixed-length vector representation, which in turn in used as the initial hidden state of a "decoder" RNN that *generates* the target sentence.

Here, we propose to follow this elegant recipe, replacing the encoder RNN by a deep convolution neural network (CNN). Over the last few years it has been convincingly shown that CNNs can produce a rich representation of the input image by embedding it to a fixed-length vector, such that this representation can be used for a variety of vision tasks [28]. Hence, it is natural to use a CNN as an image "encoder", by first pre-training it for an image classification task and using the last hidden layer as an input to the RNN decoder that generates sentences (see Fig. 1). We call this model the Neural Image Caption, or NIC.

Our contributions are as follows. First, we present an end-to-end system for the problem. It is a neural net which is fully trainable using stochastic gradient descent. Second, our model combines state-of-art sub-networks for vision and language models. These can be pre-trained on larger corpora and thus can take advantage of additional data. Finally, it yields significantly better performance compared to state-of-the-art approaches; for instance, on the Pascal dataset, NIC yielded a BLEU score of 59, to be compared to the current state-of-the-art of 25, while human performance reaches 69. On Flickr30k, we improve from 56 to 66, and on SBU, from 19 to 28.

2. Related Work

The problem of generating natural language descriptions from visual data has long been studied in computer vision, but mainly for video [7, 32]. This has led to complex systems composed of visual primitive recognizers combined with a structured formal language, e.g. And-Or Graphs or logic systems, which are further converted to natural language via rule-based systems. Such systems are heavily hand-designed, relatively brittle and have been demonstrated only on limited domains, e.g. traffic scenes or sports.

The problem of still image description with natural text has gained interest more recently. Leveraging recent advances in recognition of objects, their attributes and locations, allows us to drive natural language generation systems, though these are limited in their expressivity. Farhadi et al. [6] use detections to infer a triplet of scene elements which is converted to text using templates. Similarly, Li et al. [19] start off with detections and piece together a final description using phrases containing detected objects and relationships. A more complex graph of detections beyond triplets is used by Kulkani et al. [16], but with template-based text generation. More powerful language models based on language parsing have been used as well [23, 1, 17, 18, 5]. The above approaches have been able to describe images "in the wild", but they are heavily hand-designed and rigid when it comes to text generation.

A large body of work has addressed the problem of ranking descriptions for a given image [11, 8, 24]. Such approaches are based on the idea of co-embedding of images and text in the same vector space. For an image query, descriptions are retrieved which lie close to the image in the embedding space. Most closely, neural networks are used to co-embed images and sentences together [29] or even image crops and subsentences [13] but do not attempt to generate novel descriptions. In general, the above approaches cannot describe previously unseen compositions of objects, even though the individual objects might have been observed in the training data. Moreover, they avoid addressing the problem of evaluating how good a generated description is.

In this work we combine deep convolutional nets for image classification [12] with recurrent networks for sequence modeling [10], to create a single network that generates descriptions of images. The RNN is trained in the context of this single "end-to-end" network. The model is inspired by recent successes of sequence generation in machine translation [3, 2, 30], with the difference that instead of starting with a sentence, we provide an image processed by a convolutional net. The closest works are by Kiros et al. [15] who use a neural net, but a feedforward one, to predict the next word given the image and previous words. A recent work by Mao et al. [21] uses a recurrent NN for the same prediction task. This is very similar to the present proposal but there are a number of important differences: we use a more powerful RNN model, and provide the visual input to the RNN model directly, which makes it possible for the RNN to keep track of the objects that have been explained by the text. As a result of these seemingly insignificant differences, our system achieves substantially better results on the established benchmarks. Lastly, Kiros et al. [14] propose to construct a joint multimodal embedding space by using a powerful computer vision model and an LSTM that encodes text. In contrast to our approach, they use two separate pathways (one for images, one for text) to define a joint embedding, and, even though they can generate text, their approach is highly tuned for ranking.

3. Model

In this paper, we propose a neural and probabilistic framework to generate descriptions from images. Recent advances in statistical machine translation have shown that, given a powerful sequence model, it is possible to achieve state-of-the-art results by directly maximizing the probability of the correct translation given an input sentence in an "end-to-end" fashion – both for training and inference. These models make use of a recurrent neural network which encodes the variable length input into a fixed dimensional vector, and uses this representation to "decode" it to the desired output sentence. Thus, it is natural to use the same approach where, given an image (instead of an input sentence in the source language), one applies the same principle of "translating" it into its description.

Thus, we propose to directly maximize the probability of the correct description given the image by using the following formulation:

$$\theta^\star = \arg\max_\theta \sum_{(I,S)} \log p(S|I;\theta) \qquad (1)$$

where θ are the parameters of our model, I is an image, and

S its correct transcription. Since S represents any sentence, its length is unbounded. Thus, it is common to apply the chain rule to model the joint probability over $S_0, \ldots, S_N$, where N is the length of this particular example as

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1}) \qquad (2)$$

where we dropped the dependency on θ for convenience. At training time, (S, I) is a training example pair, and we optimize the sum of the log probabilities as described in (2) over the whole training set using stochastic gradient descent (further training details are given in Section 4).

It is natural to model $p(S_t|I, S_0, \ldots, S_{t-1})$ with a Recurrent Neural Network (RNN), where the variable number of words we condition upon up to $t-1$ is expressed by a fixed length hidden state or memory h_t. This memory is updated after seeing a new input x_t by using a non-linear function f:

$$h_{t+1} = f(h_t, x_t) \,. \qquad (3)$$

To make the above RNN more concrete two crucial design choices are to be made: what is the exact form of f and how are the images and words fed as inputs x_t. For f we use a Long-Short Term Memory (LSTM) net, which has shown state-of-the art performance on sequence tasks such as translation. This model is outlined in the next section.

For the representation of images, we use a Convolutional Neural Network (CNN). They have been widely used and studied for image tasks, and are currently state-of-the art for object recognition and detection. Our particular choice of CNN uses a novel approach to batch normalization and yields the current best performance on the ILSVRC 2014 classification competition [12]. Furthermore, they have been shown to generalize to other tasks such as scene classification by means of transfer learning [4]. The words are represented with an embedding model.

3.1. LSTM-based Sentence Generator

The choice of f in (3) is governed by its ability to deal with vanishing and exploding gradients [10], the most common challenge in designing and training RNNs. To address this challenge, a particular form of recurrent nets, called LSTM, was introduced [10] and applied with great success to translation [3, 30] and sequence generation [9].

The core of the LSTM model is a memory cell c encoding knowledge at every time step of what inputs have been observed up to this step (see Figure 2) . The behavior of the cell is controlled by "gates" – layers which are applied multiplicatively and thus can either keep a value from the gated layer if the gate is 1 or zero this value if the gate is 0. In particular, three gates are being used which control whether to forget the current cell value (forget gate f), if it should read its input (input gate i) and whether to output the new cell value (output gate o). The definition of the gates and cell update and output are as follows:

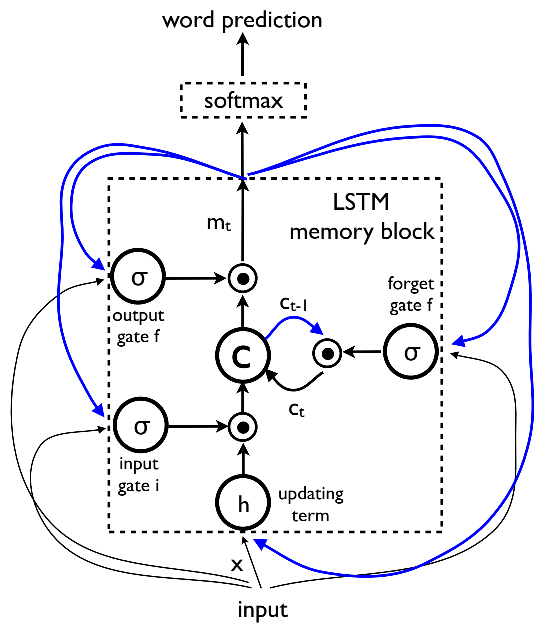

Figure 2. LSTM: the memory block contains a cell c which is controlled by three gates. In blue we show the recurrent connections – the output m at time $t-1$ is fed back to the memory at time t via the three gates; the cell value is fed back via the forget gate; the predicted word at time $t-1$ is fed back in addition to the memory output m at time t into the Softmax for word prediction.

$$\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) && (4)\\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) && (5)\\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) && (6)\\
c_t &= f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) && (7)\\
m_t &= o_t \odot c_t && (8)\\
p_{t+1} &= \text{Softmax}(m_t) && (9)
\end{aligned}$$

where $\odot$ represents the product with a gate value, and the various W matrices are trained parameters. Such multiplicative gates make it possible to train the LSTM robustly as these gates deal well with exploding and vanishing gradients [10]. The nonlinearities are sigmoid $\sigma(\cdot)$ and hyperbolic tangent $h(\cdot)$. The last equation m_t is what is used to feed to a Softmax, which will produce a probability distribution p_t over all words.

Training The LSTM model is trained to predict each word of the sentence after it has seen the image as well as all preceding words as defined by $p(S_t|I, S_0, \ldots, S_{t-1})$. For this purpose, it is instructive to think of the LSTM in unrolled form – a copy of the LSTM memory is created for the

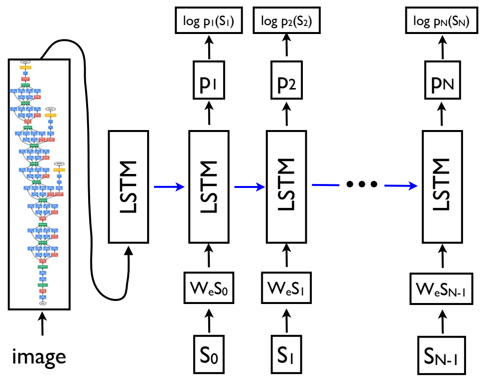

Figure 3. LSTM model combined with a CNN image embedder (as defined in [12]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters.

image and each sentence word such that all LSTMs share the same parameters and the output m_{t-1} of the LSTM at time $t-1$ is fed to the LSTM at time t (see Figure 3). All recurrent connections are transformed to feed-forward connections in the unrolled version. In more detail, if we denote by I the input image and by $S = (S_0, \ldots, S_N)$ a true sentence describing this image, the unrolling procedure reads:

$$x_{-1} = \text{CNN}(I) \tag{10}$$

$$x_t = W_e S_t, \quad t \in \{0 \ldots N-1\} \tag{11}$$

$$p_{t+1} = \text{LSTM}(x_t), \quad t \in \{0 \ldots N-1\} \tag{12}$$

where we represent each word as a one-hot vector S_t of dimension equal to the size of the dictionary. Note that we denote by S_0 a special start word and by S_N a special stop word which designates the start and end of the sentence. In particular by emitting the stop word the LSTM signals that a complete sentence has been generated. Both the image and the words are mapped to the same space, the image by using a vision CNN, the words by using word embedding W_e. The image I is only input once, at $t = -1$, to inform the LSTM about the image contents. We empirically verified that feeding the image at each time step as an extra input yields inferior results, as the network can explicitly exploit noise in the image and overfits more easily.

Our loss is the sum of the negative log likelihood of the correct word at each step as follows:

$$L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t) \; . \tag{13}$$

The above loss is minimized w.r.t. all the parameters of the LSTM, the top layer of the image embedder CNN and word embeddings W_e.

Inference There are multiple approaches that can be used to generate a sentence given an image, with NIC. The first one is **Sampling** where we just sample the first word according to p_1, then provide the corresponding embedding as input and sample p_2, continuing like this until we sample the special end-of-sentence token or some maximum length. The second one is **BeamSearch**: iteratively consider the set of the k best sentences up to time t as candidates to generate sentences of size $t + 1$, and keep only the resulting best k of them. This better approximates $S = \arg\max_{S'} p(S'|I)$. We used the BeamSearch approach in the following experiments, with a beam of size 20. Using a beam size of 1 (i.e., greedy search) did degrade our results by 2 BLEU points on average.

4. Experiments

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

4.1. Evaluation Metrics

Although it is sometimes not clear whether a description should be deemed successful or not given an image, prior art has proposed several evaluation metrics. The most reliable (but time consuming) is to ask for raters to give a subjective score on the usefulness of each description given the image. In this paper, we used this to reinforce that some of the automatic metrics indeed correlate with this subjective score, following the guidelines proposed in [11], which asks the graders to evaluate each generated sentence with a scale from 1 to 4[1].

For this metric, we set up an Amazon Mechanical Turk experiment. Each image was rated by 2 workers. The typical level of agreement between workers is 65%. In case of disagreement we simply average the scores and record the average as the score. For variance analysis, we perform bootstrapping (re-sampling the results with replacement and computing means/standard deviation over the resampled results). Like [11] we report the fraction of scores which are larger or equal than a set of predefined thresholds.

The rest of the metrics can be computed automatically assuming one has access to groundtruth, i.e. human generated descriptions. The most commonly used metric so far in the image description literature has been the BLEU score [25], which is a form of precision of word n-grams between generated and reference sentences [2]. Even though

[1] The raters are asked whether the image is described without any errors, described with minor errors, with a somewhat related description, or with an unrelated description, with a score of 4 being the best and 1 being the worst.

[2] In this literature, most previous work report BLEU-1, i.e., they only compute precision at the unigram level, whereas BLEU-n is a geometric average of precision over 1- to n-grams.

this metric has some obvious drawbacks, it has been shown to correlate well with human evaluations. In this work, we corroborate this as well, as we show in Section 4.3. An extensive evaluation protocol, as well as the generated outputs of our system, can be found at `http://nic.droppages.com/`.

Besides BLEU, one can use the perplexity of the model for a given transcription (which is closely related to our objective function in (1)). The perplexity is the geometric mean of the inverse probability for each predicted word. We used this metric to perform choices regarding model selection and hyperparameter tuning in our held-out set, but we do not report it since BLEU is always preferred [3]. A much more detailed discussion regarding metrics can be found in [31], and research groups working on this topic have been reporting other metrics which are deemed more appropriate for evaluating caption. We report two such metrics - METEOR and Cider - hoping for much more discussion and research to arise regarding the choice of metric.

Lastly, the current literature on image description has also been using the proxy task of ranking a set of available descriptions with respect to a given image (see for instance [14]). Doing so has the advantage that one can use known ranking metrics like recall@k. On the other hand, transforming the description generation task into a ranking task is unsatisfactory: as the complexity of images to describe grows, together with its dictionary, the number of possible sentences grows exponentially with the size of the dictionary, and the likelihood that a predefined sentence will fit a new image will go down unless the number of such sentences also grows exponentially, which is not realistic; not to mention the underlying computational complexity of evaluating efficiently such a large corpus of stored sentences for each image. The same argument has been used in speech recognition, where one has to produce the sentence corresponding to a given acoustic sequence; while early attempts concentrated on classification of isolated phonemes or words, state-of-the-art approaches for this task are now generative and can produce sentences from a large dictionary.

Now that our models can generate descriptions of reasonable quality, and despite the ambiguities of evaluating an image description (where there could be multiple valid descriptions not in the groundtruth) we believe we should concentrate on evaluation metrics for the generation task rather than for ranking.

4.2. Datasets

For evaluation we use a number of datasets which consist of images and sentences in English describing these images.

[3]Even though it would be more desirable, optimizing for BLEU score yields a discrete optimization problem. In general, perplexity and BLEU scores are fairly correlated.

The statistics of the datasets are as follows:

Dataset name	size		
	train	valid.	test
Pascal VOC 2008 [6]	-	-	1000
Flickr8k [26]	6000	1000	1000
Flickr30k [33]	28000	1000	1000
MSCOCO [20]	82783	40504	40775
SBU [24]	1M	-	-

With the exception of SBU, each image has been annotated by labelers with 5 sentences that are relatively visual and unbiased. SBU consists of descriptions given by image owners when they uploaded them to Flickr. As such they are not guaranteed to be visual or unbiased and thus this dataset has more noise.

The Pascal dataset is customary used for testing only after a system has been trained on different data such as any of the other four dataset. In the case of SBU, we hold out 1000 images for testing and train on the rest as used by [18]. Similarly, we reserve 4K random images from the MSCOCO validation set as test, called COCO-4k, and use it to report results in the following section.

4.3. Results

Since our model is data driven and trained end-to-end, and given the abundance of datasets, we wanted to answer questions such as "how dataset size affects generalization", "what kinds of transfer learning it would be able to achieve", and "how it would deal with weakly labeled examples". As a result, we performed experiments on five different datasets, explained in Section 4.2, which enabled us to understand our model in depth.

4.3.1 Training Details

Many of the challenges that we faced when training our models had to do with overfitting. Indeed, purely supervised approaches require large amounts of data, but the datasets that are of high quality have less than 100000 images. The task of assigning a description is strictly harder than object classification and data driven approaches have only recently become dominant thanks to datasets as large as ImageNet (with ten times more data than the datasets we described in this paper, with the exception of SBU). As a result, we believe that, even with the results we obtained which are quite good, the advantage of our method versus most current human-engineered approaches will only increase in the next few years as training set sizes will grow.

Nonetheless, we explored several techniques to deal with overfitting. The most obvious way to not overfit is to initialize the weights of the CNN component of our system to a pretrained model (e.g., on ImageNet). We did this in all the experiments (similar to [8]), and it did help quite a lot in terms of generalization. Another set of weights that could be sensibly initialized are W_e, the word embeddings. We tried initializing them from a large news corpus [22], but no significant gains were observed, and we decided to just leave them uninitialized for simplicity. Lastly, we did some model level overfitting-avoiding techniques. We tried dropout [34] and ensembling models, as well as exploring the size (i.e., capacity) of the model by trading off number of hidden units versus depth. Dropout and ensembling gave a few BLEU points improvement, and that is what we report throughout the paper.

We trained all sets of weights using stochastic gradient descent with fixed learning rate and no momentum. All weights were randomly initialized except for the CNN weights, which we left unchanged because changing them had a negative impact. We used 512 dimensions for the embeddings and the size of the LSTM memory.

Descriptions were preprocessed with basic tokenization, keeping all words that appeared at least 5 times in the training set.

4.3.2 Generation Results

We report our main results on all the relevant datasets in Tables 1 and 2. Since PASCAL does not have a training set, we used the system trained using MSCOCO (arguably the largest and highest quality dataset for this task). The state-of-the-art results for PASCAL and SBU did not use image features based on deep learning, so arguably a big improvement on those scores comes from that change alone. The Flickr datasets have been used recently [11, 21, 14], but mostly evaluated in a retrieval framework. A notable exception is [21], where they did both retrieval and generation, and which yields the best performance on the Flickr datasets up to now.

Human scores in Table 2 were computed by comparing one of the human captions against the other four. We do this for each of the five raters, and average their BLEU scores. Since this gives a slight advantage to our system, given the BLEU score is computed against five reference sentences and not four, we add back to the human scores the average difference of having five references instead of four.

Given that the field has seen significant advances in the last years, we do think it is more meaningful to report BLEU-4, which is the standard in machine translation moving forward. Additionally, we report metrics shown to correlate better with human evaluations in Table 1[4]. Despite recent efforts on better evaluation metrics [31], our model fares strongly versus human raters. However, when evaluating our captions using human raters (see Section 4.3.6), our model fares much more poorly, suggesting more work is needed towards better metrics. On the official test set for which labels are only available through the official website, our model had a 27.2 BLEU-4.

Metric	BLEU-4	METEOR	CIDER
NIC	**27.7**	**23.7**	**85.5**
Random	4.6	9.0	5.1
Nearest Neighbor	9.9	15.7	36.5
Human	21.7	25.2	85.4

Table 1. Scores on the MSCOCO development set.

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

Table 2. BLEU-1 scores. We only report previous work results when available. SOTA stands for the current state-of-the-art.

4.3.3 Transfer Learning, Data Size and Label Quality

Since we have trained many models and we have several testing sets, we wanted to study whether we could transfer a model to a different dataset, and how much the mismatch in domain would be compensated with e.g. higher quality labels or more training data.

The most obvious case for transfer learning and data size is between Flickr30k and Flickr8k. The two datasets are similarly labeled as they were created by the same group. Indeed, when training on Flickr30k (with about 4 times more training data), the results obtained are 4 BLEU points better. It is clear that in this case, we see gains by adding more training data since the whole process is data-driven and overfitting prone. MSCOCO is even bigger (5 times more training data than Flickr30k), but since the collection process was done differently, there are likely more differences in vocabulary and a larger mismatch. Indeed, all the BLEU scores degrade by 10 points. Nonetheless, the descriptions are still reasonable.

Since PASCAL has no official training set and was collected independently of Flickr and MSCOCO, we report transfer learning from MSCOCO (in Table 2). Doing transfer learning from Flickr30k yielded worse results with BLEU-1 at 53 (cf. 59).

Lastly, even though SBU has weak labeling (i.e., the labels were captions and not human generated descriptions),

[4] We used the implementation of these metrics kindly provided in http://www.mscoco.org.

[5] We computed these BLEU scores with the outputs that the authors of [14] kindly provided for their OxfordNet system.

the task is much harder with a much larger and noisier vocabulary. However, much more data is available for training. When running the MSCOCO model on SBU, our performance degrades from 28 down to 16.

4.3.4 Generation Diversity Discussion

Having trained a generative model that gives $p(S|I)$, an obvious question is whether the model generates novel captions, and whether the generated captions are both diverse and high quality. Table 3 shows some samples when returning the N-best list from our beam search decoder instead of the best hypothesis. Notice how the samples are diverse and may show different aspects from the same image. The agreement in BLEU score between the top 15 generated sentences is 58, which is similar to that of humans among them. This indicates the amount of diversity our model generates. In bold are the sentences that are not present in the training set. If we take the best candidate, the sentence is present in the training set 80% of the times. This is not too surprising given that the amount of training data is quite small, so it is relatively easy for the model to pick "exemplar" sentences and use them to generate descriptions. If we instead analyze the top 15 generated sentences, about half of the times we see a completely novel description, but still with a similar BLEU score, indicating that they are of enough quality, yet they provide a healthy diversity.

A man throwing a frisbee in a park.
A man holding a frisbee in his hand.
A man standing in the grass with a frisbee.
A close up of a sandwich on a plate.
A close up of a plate of food with french fries.
A white plate topped with a cut in half sandwich.
A display case filled with lots of donuts.
A display case filled with lots of cakes.
A bakery display case filled with lots of donuts.

Table 3. N-best examples from the MSCOCO test set. Bold lines indicate a novel sentence not present in the training set.

4.3.5 Ranking Results

While we think ranking is an unsatisfactory way to evaluate description generation from images, many papers report ranking scores, using the set of testing captions as candidates to rank given a test image. The approach that works best on these metrics (MNLM), specifically implemented a ranking-aware loss. Nevertheless, NIC is doing surprisingly well on both ranking tasks (ranking descriptions given images, and ranking images given descriptions), as can be seen in Tables 4 and 5. Note that for the Image Annotation task, we normalized our scores similar to what [21] used.

Approach	Image Annotation			Image Search		
	R@1	R@10	Med r	R@1	R@10	Med r
DeFrag [13]	13	44	14	10	43	15
m-RNN [21]	15	49	11	12	42	15
MNLM [14]	18	55	8	13	52	10
NIC	**20**	**61**	**6**	**19**	**64**	**5**

Table 4. Recall@k and median rank on Flickr8k.

Approach	Image Annotation			Image Search		
	R@1	R@10	Med r	R@1	R@10	Med r
DeFrag [13]	16	55	8	10	45	13
m-RNN [21]	18	51	10	13	42	16
MNLM [14]	**23**	**63**	**5**	**17**	**57**	**8**
NIC	17	56	7	**17**	**57**	**7**

Table 5. Recall@k and median rank on Flickr30k.

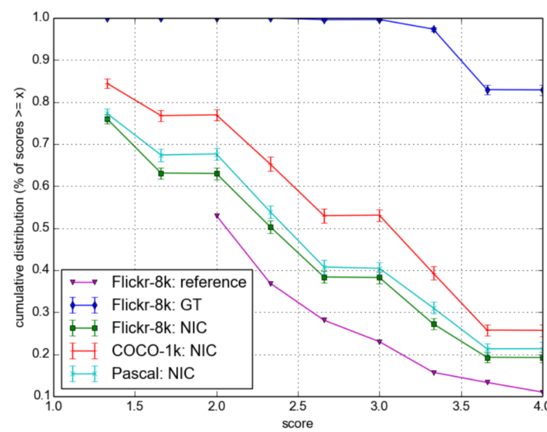

Figure 4. *Flickr-8k: NIC*: predictions produced by NIC on the Flickr8k test set (average score: 2.37); *Pascal: NIC*: (average score: 2.45); *COCO-1k: NIC*: A subset of 1000 images from the MSCOCO test set with descriptions produced by NIC (average score: 2.72); *Flickr-8k: ref*: these are results from [11] on Flickr8k rated using the same protocol, as a baseline (average score: 2.08); *Flickr-8k: GT*: we rated the groundtruth labels from Flickr8k using the same protocol. This provides us with a "calibration" of the scores (average score: 3.89)

4.3.6 Human Evaluation

Figure 4 shows the result of the human evaluations of the descriptions provided by NIC, as well as a reference system and groundtruth on various datasets. We can see that NIC is better than the reference system, but clearly worse than the groundtruth, as expected. This shows that BLEU is not a perfect metric, as it does not capture well the difference between NIC and human descriptions assessed by raters. Examples of rated images can be seen in Figure 5. It is interesting to see, for instance in the second image of the first column, how the model was able to notice the frisbee given its size.

Figure 5. A selection of evaluation results, grouped by human rating.

4.3.7 Analysis of Embeddings

In order to represent the previous word S_{t-1} as input to the decoding LSTM producing S_t, we use word embedding vectors [22], which have the advantage of being independent of the size of the dictionary (contrary to a simpler one-hot-encoding approach). Furthermore, these word embeddings can be jointly trained with the rest of the model. It is remarkable to see how the learned representations have captured some semantic from the statistics of the language. Table 4.3.7 shows, for a few example words, the nearest other words found in the learned embedding space.

Note how some of the relationships learned by the model will help the vision component. Indeed, having "horse", "pony", and "donkey" close to each other will encourage the CNN to extract features that are relevant to horse-looking animals. We hypothesize that, in the extreme case where we see very few examples of a class (e.g., "unicorn"), its proximity to other word embeddings (e.g., "horse") should provide a lot more information that would be completely lost with more traditional bag-of-words based approaches.

Word	Neighbors
car	van, cab, suv, vehicule, jeep
boy	toddler, gentleman, daughter, son
street	road, streets, highway, freeway
horse	pony, donkey, pig, goat, mule
computer	computers, pc, crt, chip, compute

Table 6. Nearest neighbors of a few example words

5. Conclusion

We have presented NIC, an end-to-end neural network system that can automatically view an image and generate a reasonable description in plain English. NIC is based on a convolution neural network that encodes an image into a compact representation, followed by a recurrent neural network that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image. Experiments on several datasets show the robustness of NIC in terms of qualitative results (the generated sentences are very reasonable) and quantitative evaluations, using either ranking metrics or BLEU, a metric used in machine translation to evaluate the quality of generated sentences. It is clear from these experiments that, as the size of the available datasets for image description increases, so will the performance of approaches like NIC. Furthermore, it will be interesting to see how one can use unsupervised data, both from images alone and text alone, to improve image description approaches.

References

[1] A. Aker and R. Gaizauskas. Generating image descriptions using dependency relational patterns. In *ACL*, 2010. 2

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014. 1, 2

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014. 1, 2, 3

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 3

[5] D. Elliott and F. Keller. Image description using visual dependency representations. In *EMNLP*, 2013. 2

[6] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010. 1, 2, 5

[7] R. Gerber and H.-H. Nagel. Knowledge representation for the generation of quantified natural language descriptions of vehicle traffic in image sequences. In *ICIP*. IEEE, 1996. 2

[8] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014. 2, 5

[9] A. Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013. 3

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8), 1997. 2, 3

[11] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 47, 2013. 2, 4, 6, 7

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *arXiv:1502.03167*, 2015. 2, 3, 4

[13] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. *NIPS*, 2014. 2, 7

[14] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *arXiv:1411.2539*, 2014. 2, 5, 6, 7

[15] R. Kiros and R. Z. R. Salakhutdinov. Multimodal neural language models. In *NIPS Deep Learning Workshop*, 2013. 2

[16] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 1, 2, 6

[17] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *ACL*, 2012. 2

[18] P. Kuznetsova, V. Ordonez, T. Berg, and Y. Choi. Treetalk: Composition and compression of trees for image descriptions. *ACL*, 2(10), 2014. 2, 5, 6

[19] 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. 2

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *arXiv:1405.0312*, 2014. 5

[21] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Explain images with multimodal recurrent neural networks. In *arXiv:1410.1090*, 2014. 2, 6, 7

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 6, 8

[23] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. C. Berg, K. Yamaguchi, T. L. Berg, K. Stratos, and H. D. III. Midge: Generating image descriptions from computer vision detections. In *EACL*, 2012. 2

[24] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2, 5, 6

[25] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: A method for automatic evaluation of machine translation. In *ACL*, 2002. 4

[26] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147, 2010. 5

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 1

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 2

[29] R. Socher, A. Karpathy, Q. V. Le, C. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. In *ACL*, 2014. 2

[30] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 1, 2, 3

[31] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. In *arXiv:1411.5726*, 2015. 5, 6

[32] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8), 2010. 2

[33] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *ACL*, 2014. 5

[34] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. In *arXiv:1409.2329*, 2014. 6