Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models

Ryan Kiros, Ruslan Salakhutdinov, Richard S. Zemel
University of Toronto
Canadian Institute for Advanced Research
{rkiros, rsalakhu, zemel}@cs.toronto.edu

Abstract

Inspired by recent advances in multimodal learning and machine translation, we introduce an encoder-decoder pipeline that learns (a): a multimodal joint embedding space with images and text and (b): a novel language model for decoding distributed representations from our space. Our pipeline effectively unifies joint image-text embedding models with multimodal neural language models. We introduce the structure-content neural language model that disentangles the structure of a sentence to its content, conditioned on representations produced by the encoder. The encoder allows one to rank images and sentences while the decoder can generate novel descriptions from scratch. Using LSTM to encode sentences, we match the state-of-the-art performance on Flickr8K and Flickr30K without using object detections. We also set new best results when using the 19-layer Oxford convolutional network. Furthermore we show that with linear encoders, the learned embedding space captures multimodal regularities in terms of vector space arithmetic e.g. *image of a blue car* - "blue" + "red" is near images of red cars. Sample captions generated for 800 images are made available for comparison.

1 Introduction

Generating descriptions for images has long been regarded as a challenging perception task integrating vision, learning and language understanding. One not only needs to correctly recognize what appears in images but also incorporate knowledge of spatial relationships and interactions between objects. Even with this information, one then needs to generate a description that is relevant and grammatically correct. With the recent advances made in deep neural networks, tasks such as object recognition and detection have made significant breakthroughs in only a short time. The task of describing images is one that now appears tractable and ripe for advancement. Being able to append large image databases with accurate descriptions for each image would significantly improve the capabilities of content-based image retrieval systems. Moreover, systems that can describe images well, could in principle, be fine-tuned to answer questions about images also.

This paper describes a new approach to the problem of image caption generation, casted into the framework of encoder-decoder models. For the encoder, we learn a joint image-sentence embedding where sentences are encoded using long short-term memory (LSTM) recurrent neural networks [1]. Image features from a deep convolutional network are projected into the embedding space of the LSTM hidden states. A pairwise ranking loss is minimized in order to learn to rank images and their descriptions. For decoding, we introduce a new neural language model called the structure-content neural language model (SC-NLM). The SC-NLM differs from existing models in that it disentangles the structure of a sentence to its content, conditioned on distributed representations produced by the encoder. We show that sampling from an SC-NLM allows us to generate realistic image captions, significantly improving over the generated captions produced by [2]. Furthermore, we argue that this combination of approaches naturally fits into the experimentation framework of [3], that is, a good encoder can be used to *rank* images and captions while a good decoder can be used to *generate* new captions from scratch. Our approach effectively unifies image-text embedding models (encoder

Figure 1: Sample generated captions. The bottom row shows different error cases. Additional results can be found at `http://www.cs.toronto.edu/~rkiros/lstm_scnlm.html`

phase) [4, 5, 6] with multimodal neural language models (decoder phase) [2] [7]. Furthermore, our method builds on analogous approaches being used in machine translation [8, 9, 10, 11].

While the application focus of our work is on image description generation and ranking, we also qualitatively analyse properties of multimodal vector spaces learned using images and sentences. We show that using a linear sentence encoder, linguistic regularities [12] also carry over to multimodal vector spaces. For example, *image of a blue car* - "blue" + "red" results in a vector that is near images of red cars. We qualitatively examine several types of analogies and structures with PCA projections. Consequently, even with a global image-sentence training objective the encoder can still be used to retrieve locally (e.g. individual words). This is analogous to pairwise ranking methods used in machine translation [13, 14].

1.1 Multimodal representation learning

A large body of work has been done on learning multimodal representations of images and text. Popular approaches include learning joint image-word embeddings [4, 5] as well as embedding images and sentences into a common space [6, 15]. Our proposed pipeline makes direct use of these ideas. Other approaches to multimodal learning include the use of deep Boltzmann machines [16], log-bilinear neural language models [2], autoencoders [17], recurrent neural networks [7] and topic-models [18]. Several bi-directional approaches to ranking images and captions have also been proposed, based off of kernel CCA [3], normalized CCA [19] and dependency tree recursive networks [6]. From an architectural standpoint, our encoder-decoder model is most similar to [20], who proposed a two-step embedding and generation procedure for semantic parsing.

1.2 Generating descriptions of images

We group together approaches to generation into three types of methods, each described here in more detail:

Template-based methods. Template-based methods involve filling in sentence templates, such as triplets, based on the results of object detections and spatial relationships [21, 22, 23, 24, 25]. While

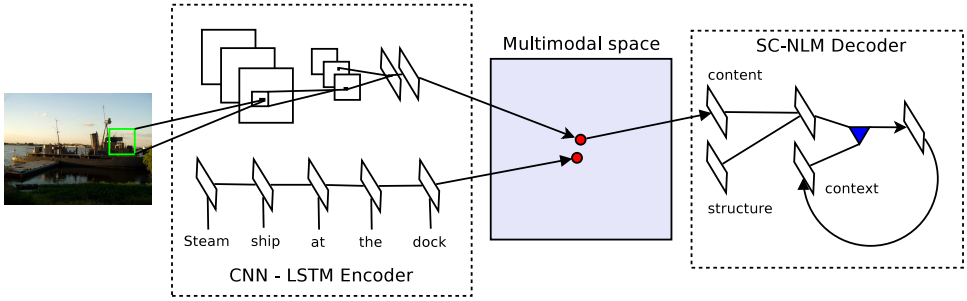


Figure 2: **Encoder:** A deep convolutional network (CNN) and long short-term memory recurrent network (LSTM) for learning a joint image-sentence embedding. **Decoder:** A new neural language model that combines structure and content vectors for generating words one at a time in sequence.

these approaches can produce accurate descriptions, they are often more 'robotic' in nature and do not generalize to the fluidity and naturalness of captions written by humans.

Composition-based methods. These approaches aim to harness existing image-caption databases by extracting components of related captions and composing them together to generate novel descriptions [26, 27]. The advantage of these approaches are that they allow for a much broader and more expressive class of captions that are more fluent and human-like then template-based approaches.

Neural network methods. These approaches aim to generate descriptions by sampling from conditional neural language models. The initial work in this area, based off of multimodal neural language models [2], generated captions by conditioning on feature vectors from the output of a deep convolutional network. These ideas were recently extended to multimodal recurrent networks with significant improvements [7]. The methods described in this paper produce descriptions that at least qualitatively on par with current state-of-the-art composition-based methods [27].

Description generation systems have been plagued with issues of evaluation. While Bleu and Rouge have been used in the past, [3] has argued that such automated evaluation methods are unreliable and do not match human judgements. These authors instead proposed that the problem of ranking images and captions can be used as a proxy for generation. Since any generation system requires a scoring function to access how well a caption and image match, optimizing this task should naturally carry over to an improvement in generation. Many recent methods have since used this approach for evaluation. None the less, the question on how to transfer improvements on ranking to generating new descriptions remained. We argue that encoder-decoder methods naturally fit into this experimentation framework. That is, the encoder gives us a way to rank images and captions and develop good scoring functions, while the decoder can use the representations learned to optimize the scoring functions as a way of generating and scoring new descriptions.

1.3 Encoder-decoder methods for machine translation

Our proposed pipeline, while new to caption generation, has already experienced several successes in Neural Machine Translation (NMT). The goal of NMT is to develop an end-to-end translation system with a large neural network, as opposed to using a neural network as an additional feature function to an existing phrase-based system. NMT methods are based on the encoder-decoder principle. That is, an encoder is used to map an English sentence to a distributed vector. A decoder is then conditioned on this vector to generate a French translation from the source text. Current methods include using a convolutional encoder and RNN decoder [8], RNN encoder and RNN decoder [9, 10] and LSTM encoder with LSTM decoder [11]. While still a young research area, these methods have already achieved performance on par with strong phrase-based systems and have improved on the start-of-the-art when used for rescoring.

We argue that it is natural to think of image caption generation as a translation problem. That is, our goal is to *translate* an image into a description. This point of view has also been used by [28] and allows us to make use of existing ideas in the machine translation literature. Furthermore, there is a natural correspondence between the concept of scoring functions (how well does a caption and image match) and alignments (which parts of a description correspond to which parts of an image) that can naturally be exploited for generating descriptions.

2 An encoder-decoder model for ranking and generation

In this section we describe our image caption generation pipeline. We first review LSTM RNNs which are used for encoding sentences, followed by how to learn multimodal distributed representations. We then review log-bilinear neural language models [29], multiplicative neural language models [30] and then introduce our structure-content neural language model.

2.1 Long short-term memory RNNs

Long short-term memory [1] is a recurrent neural network that incorporates a built in memory cell to store information and exploit long range context. LSTM memory cells are surrounded by gating units for the purpose of reading, writing and reseting information. LSTMs have been used to achieve state-of-the-art performance in several tasks such as handwriting recognition [31], sequence generation [32] speech recognition [33] and machine translation [11] among others. Dropout [34] strategies have also been proposed to prevent overfitting in deep LSTMs. [35]

Let $\mathbf{X}_t$ denote a matrix of training instances at time t. In our case, $\mathbf{X}_t$ is used to denote a matrix of word representations for the t-th word of each sentence in the training batch. Let $(\mathbf{I}_t, \mathbf{F}_t, \mathbf{C}_t, \mathbf{O}_t, \mathbf{M}_t)$ denote the input, forget, cell, output and hidden states of the LSTM at time step t. The LSTM architecture in this work is implemented using the following equations:

$$\mathbf{I}_t = \sigma(\mathbf{X}_t \cdot \mathbf{W}_{xi} + \mathbf{M}_{t-1} \cdot \mathbf{W}_{hi} + \mathbf{C}_{t-1} \cdot \mathbf{W}_{ci} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{F}_t = \sigma(\mathbf{X}_t \cdot \mathbf{W}_{xf} + \mathbf{M}_{t-1} \cdot \mathbf{W}_{hf} + \mathbf{C}_{t-1} \cdot \mathbf{W}_{cf} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{C}_t = \mathbf{F}_t \bullet \mathbf{C}_{t-1} + \mathbf{I}_t \bullet tanh(\mathbf{X}_t \cdot \mathbf{W}_{xc} + \mathbf{M}_{t-1} \cdot \mathbf{W}_{hc} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{O}_t = \sigma(\mathbf{X}_t \cdot \mathbf{W}_{xo} + \mathbf{M}_{t-1} \cdot \mathbf{W}_{ho} + \mathbf{C}_t \cdot \mathbf{W}_{co} + \mathbf{b}_o) \quad (4)$$

$$\mathbf{M}_t = \mathbf{O}_t \bullet tanh(\mathbf{C}_t) \quad (5)$$

where (σ) denotes the sigmoid activation function, $(\cdot)$ indicates matrix multiplication and $(\bullet)$ indicates component-wise multiplication. [1]

2.2 Multimodal distributed representations

Suppose for training we are given image-description pairs each corresponding to an image and a description that correctly describes the image. Images are represented as the top layer (before the softmax) of a convolutional network trained on the ImageNet classification task [36].

Let D be the dimensionality of an image feature vector (e.g. 4096 for AlexNet [36]), K the dimensionality of the embedding space and let V be the number of words in the vocabulary. Let $\mathbf{W}_I \in \mathbb{R}^{K \times D}$ and $\mathbf{W}_T \in \mathbb{R}^{K \times V}$ be the image embedding matrix and word embedding matrices, respectively. Given an image description $S = \{w_1, \ldots, w_N\}$ with words $w_1, \ldots, w_N$, [2] let $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\}, \mathbf{w}_i \in \mathbb{R}^K, i = 1, \ldots, n$ denote the corresponding word representations to words $w_1, \ldots, w_N$ (entries in the matrix $\mathbf{W}_T$). The representation of a sentence $\mathbf{v}$ is the hidden state of the LSTM at time step N (i.e. the vector $\mathbf{m}_t$). We note that other approaches for computing sentence representations for image-text embeddings have been proposed, including dependency tree RNNs [6] and bags of dependency parses [15]. Let $\mathbf{q} \in \mathbb{R}^D$ denote an image feature vector (for the image corresponding to description S) and let $\mathbf{x} = \mathbf{W}_I \cdot \mathbf{q} \in \mathbb{R}^K$ be the image embedding. We define a scoring function $s(\mathbf{x}, \mathbf{v}) = \mathbf{x} \cdot \mathbf{v}$, where $\mathbf{x}$ and $\mathbf{v}$ are first scaled to have unit norm (making s equivalent to cosine similarity). Let $\boldsymbol{\theta}$ denote all the parameters to be learned ($\mathbf{W}_I$ and all the LSTM weights) [3]. We optimize the following pairwise ranking loss:

$$\min_{\boldsymbol{\theta}} \sum_{\mathbf{x}} \sum_{k} \max\{0, \alpha - s(\mathbf{x}, \mathbf{v}) + s(\mathbf{x}, \mathbf{v}_k)\} + \sum_{\mathbf{v}} \sum_{k} \max\{0, \alpha - s(\mathbf{v}, \mathbf{x}) + s(\mathbf{v}, \mathbf{x}_k)\} \quad (6)$$

where $\mathbf{v}_k$ is a contrastive (non-descriptive) sentence for image embedding $\mathbf{x}$, and vice-versa with $\mathbf{x}_k$. For all of our experiments, we initialize the word embeddings $\mathbf{W}_T$ to be pre-computed $K = 300$ dimensional vectors learned using a continuous bag-of-words model [37]. The contrastive terms are chosen randomly from the training set and resampled every epoch.

[1] For additional details on LSTM: `http://people.idsia.ch/~juergen/rnn.html`.

[2] As a slight abuse of notation, we refer to w_i as both a word and an index into the word embedding matrix.

[3] We keep the word embedding matrix $\mathbf{W}_T$ fixed.

2.3 Log-bilinear neural language models

The log-bilinear language model (LBL) [29] is a deterministic model that may be viewed as a feed-forward neural network with a single linear hidden layer. Each word w in the vocabulary is represented as a K-dimensional real-valued vector $\mathbf{w} \in \mathbb{R}^K$, as in the case of the encoder. Let $\mathbf{R}$ denote a $V \times K$ matrix of word representation vectors [4] where V is the vocabulary size. Let $(w_1, \dots w_{n-1})$ be a tuple of $n-1$ words where $n-1$ is the context size. The LBL model makes a linear prediction of the next word representation as

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{w}_i, \tag{7}$$

where $\mathbf{C}^{(i)}, i = 1, \dots, n-1$ are $K \times K$ context parameter matrices. Thus, $\hat{\mathbf{r}}$ is the predicted representation of $\mathbf{w}_n$. The conditional probability $P(w_n = i | w_{1:n-1})$ of w_n given $w_1, \dots, w_{n-1}$ is

$$P(w_n = i | w_{1:n-1}) = \frac{\exp(\hat{\mathbf{r}}^T \mathbf{r}_i + b_i)}{\sum_{j=1}^{V} \exp(\hat{\mathbf{r}}^T \mathbf{r}_j + b_j)}, \tag{8}$$

where $\mathbf{b} \in \mathbb{R}^V$ is a bias vector. Learning is done with stochastic gradient descent.

2.4 Multiplicative neural language models

Suppose now we are given a vector $\mathbf{u} \in \mathbb{R}^K$ from the multimodal vector space, which has an association with a word sequence $S = \{w_1, \dots, w_N\}$. For example, $\mathbf{u}$ may be the embedded representation of an image whose description is given by S. A multiplicative neural language model [30] models the distribution $P(w_n = i | w_{1:n-1}, \mathbf{u})$ of a new word w_n given context from the previous words and the vector $\mathbf{u}$. A multiplicative model has the additional property that the word embedding matrix is instead replaced with a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{V \times K \times G}$ where G is the number of slices. Given $\mathbf{u}$, we can compute a word representation matrix as a function of $\mathbf{u}$ as $\boldsymbol{\mathcal{T}}^u = \sum_{i=1}^{G} u_i \boldsymbol{\mathcal{T}}^{(i)}$ i.e. word representations with respect to $\mathbf{u}$ are computed as a linear combination of slices weighted by each component u_i of $\mathbf{u}$. Here, the number of slices G is equal to K, the dimensionality of $\mathbf{u}$.

It is often unnecessary to use a fully unfactored tensor. As in e.g. [38, 39], we re-represent $\boldsymbol{\mathcal{T}}$ in terms of three matrices $\mathbf{W}^{fk} \in \mathbb{R}^{F \times K}$, $\mathbf{W}^{fd} \in \mathbb{R}^{F \times G}$ and $\mathbf{W}^{fv} \in \mathbb{R}^{F \times V}$, such that

$$\boldsymbol{\mathcal{T}}^u = (\mathbf{W}^{fv})^\top \cdot \text{diag}(\mathbf{W}^{fd} \mathbf{u}) \cdot \mathbf{W}^{fk} \tag{9}$$

where $\text{diag}(\cdot)$ denotes the matrix with its argument on the diagonal. These matrices are parametrized by a pre-chosen number of factors F. In [30], the conditioning vector $\mathbf{u}$ is referred to as an *attribute* and using a third-order model of words allows one to model conditional similarity: how meanings of words change as a function of the attributes they're conditioned on.

Let $\mathbf{E} = (\mathbf{W}^{fk})^\top \mathbf{W}^{fv}$ denote a 'folded' $K \times V$ matrix of word embeddings. Given the context $w_1, \dots, w_{n-1}$, the predicted next word representation $\hat{\mathbf{r}}$ is given by:

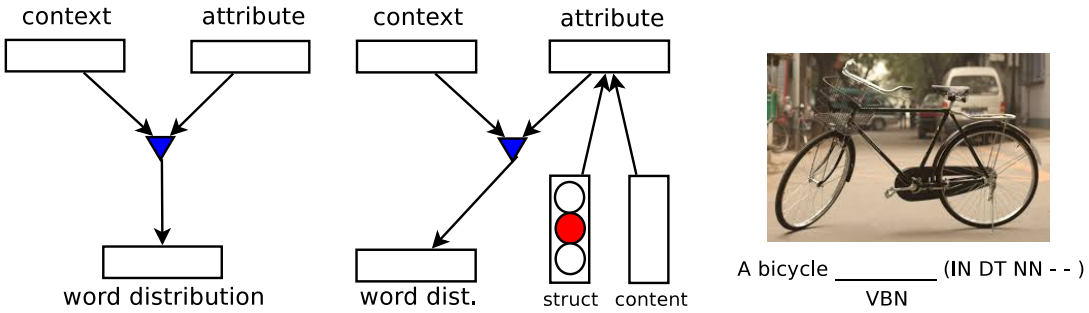
$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{E}(:, w_i), \tag{10}$$

where $\mathbf{E}(:, w_i)$ denotes the column of $\mathbf{E}$ for the word representation of w_i and $\mathbf{C}^{(i)}, i = 1, \dots, n-1$ are $K \times K$ context matrices. Given a predicted next word representation $\hat{\mathbf{r}}$, the factor outputs are $\mathbf{f} = (\mathbf{W}^{fk} \hat{\mathbf{r}}) \bullet (\mathbf{W}^{fd} \mathbf{u})$, where $\bullet$ is a component-wise product. The conditional probability $P(w_n = i | w_{1:n-1}, \mathbf{u})$ of w_n given $w_1, \dots, w_{n-1}$ and $\mathbf{u}$ can be written as

$$P(w_n = i | w_{1:n-1}, \mathbf{u}) = \frac{\exp\big((\mathbf{W}^{fv}(:, i))^\top \mathbf{f} + b_i\big)}{\sum_{j=1}^{V} \exp\big((\mathbf{W}^{fv}(:, j))^\top \mathbf{f} + b_j\big)},$$

where $\mathbf{W}^{fv}(:, i)$ denotes the column of $\mathbf{W}^{fv}$ corresponding to word i. In contrast to the log-bilinear model, the matrix of word representations $\mathbf{R}$ from before is replaced with the factored tensor $\boldsymbol{\mathcal{T}}$ that we have derived. We compared the multiplicative model against an additive variant [2] and found on large datasets, such as the SBU Captioned Photo dataset [40], the multiplicative variant significantly outperforms its additive counterpart. Thus, the SC-NLM is derived from the multiplicative variant.

[4] Note that this is a different matrix then that used by the encoder. We use the same vocabulary throughout both models.



(a) Multiplicative NLM (b) Structure-content NLM (c) SC-NLM prediction

Figure 3: Left: multiplicative neural language model. Middle: Structure-content neural language model (SC-NLM). Right: The prediction problem of an SC-NLM.

2.5 Structure-content neural language models

We now describe the structure-content neural language model. Suppose that, along with a description $S = \{w_1, \ldots, w_N\}$, we are also given a sequence of word-specific structure variables $T = \{t_1, \ldots, t_N\}$. Throughout our experiments, each t_i corresponds to the part-of-speech for word w_i, although other possibilities can be used instead. Given an embedding $\mathbf{u}$ (the content vector), our goal is to model the distribution $P(w_n = i | w_{1:n-1}, t_{n:n+k}, \mathbf{u})$ from previous word context $w_{1:n-1}$ and forward structure context $t_{n:n+k}$, where k is the forward context size. Figure 3 gives an illustration of the model and prediction problem. Intuitively, the structure variables help guide the model during the generation phrase and can be thought of as a soft template to help avoid the model from generating grammatical nonsense. Note that this model shares a resemblance with the NNJM of [41] for machine translation, where the previous word context are predicted words in the target language, and the forward context are words in the source language.

Our model can be interpreted as a multiplicative neural language model but where the attribute vector is no longer $\mathbf{u}$ but instead an additive function of $\mathbf{u}$ and the structure variables T. Let $\{\mathbf{t}_n, \ldots, \mathbf{t}_{n+k}\}, \mathbf{t}_i \in \mathbb{R}^K, i = n, \ldots, n+k$ be embedding vectors for the structure variables T. These are obtained from a learned lookup table in the same way as words are. We introduce a sequence of $G \times G$ structure context matrices $\mathbf{T}^{(i)}, i = n, \ldots, n+k$ which play the same role as the word context matrices $\mathbf{C}^{(i)}$. Let $\mathbf{T}_u$ denote a $G \times K$ context matrix for the multimodal vector $\mathbf{u}$. The attribute vector $\hat{\mathbf{u}}$ of combined structure and content information is computed as

$$\hat{\mathbf{u}} = \left[\left(\sum_{i=n}^{n+k} \mathbf{T}^{(i)} \mathbf{t}_i \right) + \mathbf{T}^{(u)} \mathbf{u} + \mathbf{b} \right]_+ \tag{11}$$

where $[\cdot]_+ = max\{\cdot, 0\}$ is a ReLU non-linearity and $\mathbf{b}$ is a bias vector. The vector $\hat{\mathbf{u}}$ now plays the same role as the vector $\mathbf{u}$ for the multiplicative model previously described and the remainder of the model remains unchanged. Our experiments use $G = K = 300$ and factors $F = 100$.

The SC-NLM is trained on a large collection of image descriptions (e.g. Flickr30K). There are several choices available for representing the conditioning vectors $\mathbf{u}$. One choice would be to use the embedding of the corresponding image. An alternative choice, which is the approach we take, is to condition on the embedding vector for the description S computed with the LSTM. The advantage of this approach is that the SC-NLM can be trained purely on text alone. This allows us to make use of large amounts of monolingual text (e.g. non image captions) to improve the quality of the language model. Since the embedding vectors of S share a joint space with the image embeddings, we can also condition the SC-NLM on image embeddings (e.g. at test time, when no description is available) after the model has been trained. This is a significant advantage over a conditional language model that explicitly requires image-caption pairs for training and highlights the strength of a multimodal encoding space.

Due to space limitations, we leave the full details of our caption generation procedure to the supplementary material.

Flickr8K								
	Image Annotation				Image Search			
Model	**R@1**	**R@5**	**R@10**	**Med *r***	**R@1**	**R@5**	**R@10**	**Med *r***
Random Ranking	0.1	0.6	1.1	631	0.1	0.5	1.0	500
SDT-RNN [6]	4.5	18.0	28.6	32	6.1	18.5	29.0	29
† DeViSE [5]	4.8	16.5	27.3	28	5.9	20.1	29.6	29
† SDT-RNN [6]	6.0	22.7	34.0	23	6.6	21.6	31.7	25
DeFrag [15]	5.9	19.2	27.3	34	5.2	17.6	26.5	32
† DeFrag [15]	12.6	32.9	44.0	14	9.7	29.6	42.5	15
m-RNN [7]	14.5	37.2	48.5	11	11.5	31.0	42.4	15
Our model	13.5	36.2	45.7	13	10.4	31.0	43.7	14
Our model (OxfordNet)	**18.0**	**40.9**	**55.0**	**8**	**12.5**	**37.0**	**51.5**	**10**

Table 1: Flickr8K experiments. **R@K** is Recall@K (high is good). **Med *r*** is the median rank (low is good). Best results overall are **bold** while best results without OxfordNet features are underlined. A † infront of the method indicates that object detections were used along with single frame features.

3 Experiments

3.1 Image-sentence ranking

Our main quantitative results is to establish the effectiveness of using an LSTM sentence encoder for ranking image and descriptions. We perform the same experimental procedure as done by [15] on the Flickr8K [3] and Flickr30K [42] datasets. These datasets come with 8,000 and 30,000 images respectively with each image annotated using 5 sentences by independent annotators. As with [15], we did not do any explicit text preprocessing. We used two convolutional network architectures for extracting 4096 dimensional image features: the Toronto ConvNet [5] as well as the 19-layer OxfordNet [43] which finished 2nd place in the ILSVRC 2014 classification competition. Following the protocol of [15], 1000 images are used for validation, 1000 for testing and the rest are used for training. Evaluation is performed using Recall@K, namely the mean number of images for which the correct caption is ranked within the top-K retrieved results (and vice-versa for sentences). We also report the median rank of the closest ground truth result from the ranked list. We compare our results to each of the following methods:

DeViSE. The deep visual semantic embedding model [5] was proposed as a way of performing zero-shot object recognition and was used as a baseline by [15]. In this model, sentences are represented as the mean of their word embeddings and the objective function optimized matches ours.

SDT-RNN. The semantic dependency tree recursive neural network [6] is used to learn sentence representations for embedding into a joint image-sentence space. The same objective is used.

DeFrag. Deep fragment embeddings [15] were proposed as an alternative to embedding full-frame image features and take advantage of object detections from the R-CNN [44] detector. Descriptions are represented as a bag of dependency parses. Their objective incorporates both a global and fragment objectives, for which their global objective matches ours.

m-RNN. The multimodal recurrent neural network [7] is a recently proposed method that uses perplexity as a bridge between modalities, as first introduced by [2]. Unlike all other methods, the m-RNN does not use a ranking loss and instead optimizes the log-likelihood of predicting the next word in a sequence conditioned on an image.

Our LSTMs use 1 layer with 300 units and weights initialized uniformly from [-0.08, 0.08]. The margin α was set to $\alpha = 0.2$, which we found performed well on both datasets. Training is done using stochastic gradient descent with an initial learning rate of 1 and was exponentially decreased. We used minibatch sizes of 40 on Flickr8K and 100 on Flickr30K. No momentum was used. The same hyperparameters are used for the OxfordNet experiments.

3.1.1 Results

Tables 1 and 2 illustrate our results on Flickr8K and Flickr30K respectively. The performance of our model is comparable to that of the m-RNN. For some metrics we outperform or match existing results while on others m-RNN outperforms our model. The m-RNN does not learn an explicit embedding between images and sentences and relies on perplexity as a means of retrieval. Methods that

[5] https://github.com/TorontoDeepLearning/convnet

Flickr30K								
	Image Annotation				Image Search			
Model	**R@1**	**R@5**	**R@10**	**Med *r***	**R@1**	**R@5**	**R@10**	**Med *r***
Random Ranking	0.1	0.6	1.1	631	0.1	0.5	1.0	500
† DeViSE [5]	4.5	18.1	29.2	26	6.7	21.9	32.7	25
† SDT-RNN [6]	9.6	29.8	41.1	16	8.9	29.8	41.1	16
† DeFrag [15]	14.2	37.7	51.3	10	10.2	30.8	44.2	14
† DeFrag + Finetune CNN [15]	16.4	40.2	54.7	8	10.3	31.4	44.5	13
m-RNN [7]	18.4	40.2	50.9	10	12.6	31.2	41.5	16
Our model	14.8	39.2	50.9	10	11.8	34.0	46.3	13
Our model (OxfordNet)	**23.0**	**50.7**	**62.9**	**5**	**16.8**	**42.0**	**56.5**	**8**

Table 2: Flickr30K experiments. **R@K** is Recall@K (high is good). **Med *r*** is the median rank (low is good). Best results overall are **bold** while best results without OxfordNet features are underlined. A † infront of the method indicates that object detections were used along with single frame features.

learn explicit embedding spaces have a significant speed advantage over perplexity-based retrieval methods, since retrieval is easily done with a single matrix multiply of stored embedding vectors from the dataset with the query vector. Thus explicit embedding methods are much better suited for scaling to large datasets.

Perhaps more interestingly is the fact that both our method and the m-RNN outperform existing models that integrate object detections. This is contradictory to [6], where recurrent networks are the worst performing models. This highlights the effectiveness of LSTM cells for encoding dependencies across descriptions and learning meaningful distributed sentence representations. Integrating object detections into our framework should almost surely improve performance as well as allow for interpretable retrievals, as in the case of DeFrag.

Using image features from the OxfordNet model results in a significant performance boost across all metrics, giving new state-of-the-art numbers on these evaluation tasks.

3.2 Multimodal linguistic regularities

Word embeddings learned with skip-gram [37] or neural language models [45] were shown by [12] to exhibit linguistic regularities that allow these models to perform analogical reasoning. For instance, "man" is to "woman" as "king" is to ? can be answered by finding the closest vector to "king" - "man" + "woman". A natural question we ask is whether multimodal vector spaces exhibit the same phenomenon. Would *image of a blue car* - "blue" + "red" be near images of red cars?

Suppose that we train an embedding model with a linear encoder, namely $\mathbf{v} = \sum_{i=1}^{N} \mathbf{w}_i$ for word vectors $\mathbf{w}_i$ and sentence vector $\mathbf{v}$ (where both $\mathbf{v}$ and the image embedding are normalized to unit length). Using our example above, let $\mathbf{v}_{blue}$, $\mathbf{v}_{red}$ and $\mathbf{v}_{car}$ denote the word embeddings for blue, red and car respectively. Let $\mathbf{I}_{bcar}$ and $\mathbf{I}_{rcar}$ denote embeddings of images with blue and red cars. After training a linear encoder, the model has the property that $\mathbf{v}_{blue} + \mathbf{v}_{car} \approx \mathbf{I}_{bcar}$ and $\mathbf{v}_{red} + \mathbf{v}_{car} \approx \mathbf{I}_{rcar}$. It follows that

$$\mathbf{v}_{car} \approx \mathbf{I}_{bcar} - \mathbf{v}_{blue} \tag{12}$$

$$\mathbf{v}_{red} + \mathbf{v}_{car} \approx \mathbf{I}_{bcar} - \mathbf{v}_{blue} + \mathbf{v}_{red} \tag{13}$$

$$\mathbf{I}_{rcar} \approx \mathbf{I}_{bcar} - \mathbf{v}_{blue} + \mathbf{v}_{red} \tag{14}$$

Thus given a query image $\mathbf{q}$, a negative word $\mathbf{w}_n$ and a positive word $\mathbf{w}_p$ (all with unit norm), we seek an image $\mathbf{x}^*$ such that:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \frac{(\mathbf{q} - \mathbf{w}_n + \mathbf{w}_p)^\top \mathbf{x}}{\|\mathbf{q} - \mathbf{w}_n + \mathbf{w}_p\|} \tag{15}$$

The supplementary material contains qualitative evidence that the above holds for several types of regularities and images. [6] In our examples, we consider retrieving the top-4 nearest images. Occasionally we observed that a poor result would be obtained within the top-4 among good results. We found a simple strategy for removing these cases is to first retrieve the top N nearest images, then re-sort these based on their distance to the mean of the N images.

It is worth noting that these kinds of regularities are not well observed with an LSTM encoder, since sentences are no longer just a sum of their words. The linear encoder is roughly equivalent to the

[6] For this model we finetune the word representations.

DeViSE baselines in tables 1 and 2, which perform significantly worse for retrieval than an LSTM encoder. So while these regularities are interesting the learned multimodal vector space is not well apt for ranking sentences and images.

3.3 Image caption generation

We generated image descriptions for roughly 800 images from the SBU captioned photo dataset [40]. These are the same images used to display results by the current state-of-the-art composition based approach, TreeTalk [27]. [7] Our LSTM encoder and SC-NLM decoder were trained by concatenating the Flickr30K dataset with the recently released Microsoft COCO dataset [46], which combined give us over 100,000 images and over 500,000 descriptions for training. The SBU dataset contains 1 million images each with a single description and was used by [27] for training their model. While the SBU dataset is larger, the annotated descriptions are noisier and more personalized.

The generated results can be found at `http://www.cs.toronto.edu/~rkiros/lstm_scnlm.html` [8]. For each image we show the original caption, the nearest neighbour sentence from the training set, the top-5 generated samples from our model and the best generated result from TreeTalk. The nearest neighbour sentence is displayed to demonstrate that our model has not simply learned to copy the training data. Our generated descriptions are arguably the nicest ones to date.

4 Discussion

When generating a description, it is often the case that only a small region is relevant at any given time. We are developing an attention-based model that jointly learns to align parts of captions to images and use these alignments to determine where to attend next, thus dynamically modifying the vectors used for conditioning the decoder. We also plan on experimenting with LSTM decoders as well as deep and bidirectional LSTM encoders.

Acknowledgments

We would like to thank Nitish Srivastava for assistance with his ConvNet package as well as preparing the Oxford convolutional network. We also thank the anonymous reviewers from the NIPS 2014 deep learning workshop for their comments and suggestions.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[2] Ryan Kiros, Richard S Zemel, and Ruslan Salakhutdinov. Multimodal neural language models. *ICML*, 2014.

[3] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 2013.

[4] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 2010.

[5] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeffrey Dean, and Tomas Mikolov MarcAurelio Ranzato. Devise: A deep visual-semantic embedding model. *NIPS*, 2013.

[6] Richard Socher, Q Le, C Manning, and A Ng. Grounded compositional semantics for finding and describing images with sentences. In *TACL*, 2014.

[7] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

[8] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *EMNLP*, 2013.

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*, 2014.

[7] `http://ilp-cky.appspot.com/generation`

[8] These results use features from the Toronto ConvNet.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *NIPS*, 2014.

[12] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *NAACL-HLT*, 2013.

[13] Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. *ICLR*, 2014.

[14] Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributional semantics. In *ACL*, 2014.

[15] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. *NIPS*, 2014.

[16] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, 2012.

[17] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Ng. Multimodal deep learning. In *ICML*, 2011.

[18] Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Learning cross-modality similarity for multinomial data. In *ICCV*, 2011.

[19] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*. 2014.

[20] Phil Blunsom, Nando de Freitas, Edward Grefenstette, Karl Moritz Hermann, et al. A deep architecture for semantic parsing. In *ACL 2014 Workshop on Semantic Parsing*, 2014.

[21] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011.

[22] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*. 2010.

[23] Siming Li, Girish Kulkarni, Tamara L Berg, Alexander C Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *CONLL*, 2011.

[24] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[25] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *EACL*, 2012.

[26] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. ACL, 2012.

[27] Polina Kuznetsova, Vicente Ordonez, Tamara L. Berg, and Yejin Choi. Treetalk : Composition and compression of trees for image descriptions. *TACL*, 2014.

[28] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[29] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *ICML*, pages 641–648, 2007.

[30] Ryan Kiros, Richard S Zemel, and Ruslan Salakhutdinov. A multiplicative model for learning distributed text-based attribute representations. *NIPS*, 2014.

[31] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *TPAMI*, 2009.

[32] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[33] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *IEEE Workshop on ASRU*, 2013.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[35] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[38] Roland Memisevic and Geoffrey Hinton. Unsupervised learning of image transformations. In *CVPR*, pages 1–8, 2007.

[39] Alex Krizhevsky, Geoffrey E Hinton, et al. Factored 3-way restricted boltzmann machines for modeling natural images. In *AISTATS*, pages 621–628, 2010.

[40] Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[41] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. *ACL*, 2014.

[42] Peter Young Alice Lai Micah Hodosh and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.

[45] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *JMLR*, 2003.

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2014.

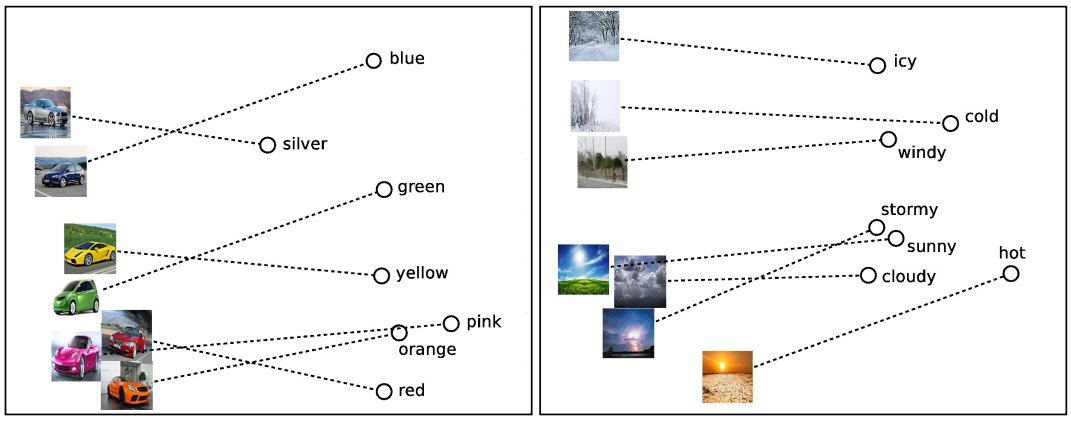
5 Supplementary material: Additional experimentation and details

5.1 Multimodal linguistic regularities

(a) Simple cases (b) Colors

(c) Image structure (d) Sanity check

Figure 4: Multimodal vector space arithmetic. Query images were downloaded online and retrieved images are from the SBU dataset.



(a) Colors (b) Weather

Figure 5: PCA projection of the 300-dimensional word and image representations for (a) cars and colors and (b) weather and temperature.

Figure 4 illustrates sample results using a model trained on the SBU dataset. All queries were downloaded online and retrieved images are from the SBU images used for training. What is of interest to note is that the resulting images depend highly on the image used for the query. For example, searching for the word 'night' retrieves arbitrary images taken at night. On the other hand, an image with a building predominantly as its focus will return night images when 'day' is

subtracted and 'night' is added. A similar phenomenon occurs with the example of cats, bowls and boxes. As additional visualizations, we computed PCA projections of cars and their corresponding colors as well as images and the weather occurrences in Figure 5. These results give us strong evidence for the regularities apparent in multimodal vector spaces trained with linear encoders. Of course, sensible results are only likely to be obtained if (a) the content of the image is correctly recognized, (b) the subtraction word is relevant to the image and (c) an image exists that is sensible for the corresponding query.

5.2 Image description generation

The SC-NLM was trained on the concatenation of training sentences from both Flickr30K and Microsoft COCO. Given an image, we first map it into the multimodal space. From this embedding, we define 2 sets of candidate conditioning vectors to the SC-NLM:

Image embedding. The embedded image itself. Note that the SC-NLM was not trained with images but can be conditioned on images since the embedding space is multimodal.

top-N nearest words and sentences. After first computing the image embedding, we obtain the top-N nearest neighbour words and training sentences using cosine similarity. These retrievals are treated as a 'bag of concepts' for which we compute an embedding vector as the mean of each concept. All of our results use $N = 5$.

Along with the candidate conditioning vectors, we also compute candidate POS sequences used by the SC-NLM. For this, we obtain a set of all POS sequences from the training set whose lengths were between 4 and 12, inclusive. Captions are generated by first sampling a conditioning vector, next sampling a POS sequence, then computing a MAP estimate from the SC-NLM. We generate a large list of candidate descriptions (1000 for each image in our results) and rank these candidates using a scoring function. Our scoring function consists of two feature functions:

Translation model. The candidate description is embedded into the multimodal space using the LSTM. We then compute a translation score as the cosine similarity between the image embedding and the embedding of the candidate description. This scores how relevant the content of the candidate is to the image. We also augment to this score a multiplicative penalty to non-stopwords that appear too frequently in the description. [9]

Language model. We trained a Kneser-Ney trigram model on a large corpus and compute the log-probability of the candidate under the model. This scores how reasonable of an English sentence is the candidate.

The total score of a caption is then the weighted sum of the translation and language models. Due to the challenge of quantitatively evaluating generated descriptions, we tuned the weights by hand on qualitative results alone. All of the candidate descriptions are ranked by their scores, and the top-5 captions are returned.

[9] For instance, given an image of a car, we would want a candidate to be ranked low if each noun in the description was 'car'.