Boosting Image Captioning with Attributes*

Ting Yao [†], Yingwei Pan [‡], Yehao Li [§], Zhaofan Qiu [‡], and Tao Mei [†]
[†] Microsoft Research, Beijing, China
[‡] University of Science and Technology of China, Hefei, China
[§] Sun Yat-Sen University, Guangzhou, China
{tiyao, tmei}@microsoft.com, {panyw.ustc, yehaoli.sysu, zhaofanqiu}@gmail.com

Abstract

Automatically describing an image with a natural language has been an emerging challenge in both fields of computer vision and natural language processing. In this paper, we present Long Short-Term Memory with Attributes (LSTM-A) - a novel architecture that integrates attributes into the successful Convolutional Neural Networks (CNNs) plus Recurrent Neural Networks (RNNs) image captioning framework, by training them in an end-to-end manner. Particularly, the learning of attributes is strengthened by integrating inter-attribute correlations into Multiple Instance Learning (MIL). To incorporate attributes into captioning, we construct variants of architectures by feeding image representations and attributes into RNNs in different ways to explore the mutual but also fuzzy relationship between them. Extensive experiments are conducted on COCO image captioning dataset and our framework shows clear improvements when compared to state-of-the-art deep models. More remarkably, we obtain METEOR/CIDEr-D of 25.5%/100.2% on testing data of widely used and publicly available splits in [10] when extracting image representations by GoogleNet and achieve superior performance on COCO captioning Leaderboard.

1. Introduction

Accelerated by tremendous increase in Internet bandwidth and proliferation of sensor-rich mobile devices, image data has been generated, published and spread explosively, becoming an indispensable part of today's big data. This has encouraged the development of advanced techniques for a broad range of image understanding applications. A fundamental issue that underlies the success of these technological advances is the recognition [8, 26, 28]. Recently, researchers have strived to automatically describe the content of an image with a complete and natural sentence, which has a great potential impact for instance on robotic vision or helping visually impaired people. Nevertheless, this problem is very challenging, as description generation model should capture not only the objects/scenes presented in the image, but also be capable of expressing how the objects/scenes relate to each other in a nature sentence.

The main inspiration of recent attempts on this problem [5, 31, 33] are from the advances by using RNNs in machine translation [27], which is to translate a text from one language (e.g., English) to another (e.g., Chinese). The basic idea is to perform a sequence to sequence learning for translation, where an encoder RNN reads the input sequential sentence, one word at a time till the end of the sentence and then a decoder RNN is exploited to generate the sentence in target language, one word at each time step. Following this philosophy, it is natural to employ a CNN instead of the encoder RNN for image captioning, which is regarded as an image encoder to produce image representations.

While encouraging performances are reported, these CNN plus RNN image captioning methods translate directly from image representations to language, without explicitly taking more high-level semantic information from images into account. On the other hand, attributes are properties observed in images with rich semantic cues and have been proved to be effective in visual recognition [23]. Therefore, a valid question is how to incorporate high-level image attributes into CNN plus RNN image captioning architecture as complementary knowledge in addition to image representations. We investigate particularly in this paper the architectures by exploiting the mutual relationship between image representations and attributes for enhancing image description generation. Specifically, to better demonstrate the impact of simultaneously utilizing the two kinds of representations, we devise variants of architectures by feeding them into RNN in different placements and moments, e.g., leveraging only attributes, inserting image representations first and then attributes or vice versa, and inputting image representations/attributes once or at each time step. More-

*This work was performed when Yingwei Pan, Yehao Li and Zhaofan Qiu were visiting Microsoft Research as research interns.

over, considering attributes are vital to our proposal, we endow the Multiple Instance Learning (MIL) framework with more power of exploring inter-attribute correlations.

The main contribution of this work is the proposal of attribute augmented architectures by integrating the attributes into CNN plus RNN image captioning framework, which is a problem not yet fully understood in the literature. By leveraging more knowledge for building richer representations and description models, our work takes a further step forward to enhance image captioning. More importantly, the utilization of attributes also has a great potential to be an elegant solution of generating open-vocabulary sentences, making image captioning system really practical.

2. Related Work

The research on image captioning has proceeded along three different dimensions: template-based methods [11, 18, 34], search-based approaches [4, 7], and language-based models [5, 16, 31, 32, 33, 36, 38].

The template-based methods firstly align each sentence fragments (e.g., subject, verb, object) with detected words from image content and then generate the sentence with predefined language templates. Obviously, most of them highly depend on the templates of sentence and always generate sentences with syntactical structure. For example, Kulkarni *et al.* employ Conditional Random Field (CRF) model to predict labeling based on the detected objects, attributes and prepositions, and then generate sentence with a template by filling in slots with the most likely labeling [11]. Similarly, Yang *et al.* utilize HMM to select the best objects, scenes, verbs, and prepositions with the highest log-likelihood ratio for template-based sentence generation in [34].

Search-based approaches "generate" sentence for an image by selecting the most semantically similar sentences from sentence pool or directly copying sentences from other visually similar images. This direction indeed can achieve human-level descriptions as all sentences are from existing human-generated sentences. The need to collect human-generated sentences, however, makes the sentence pool hard to be scaled up. Moreover, the approaches in this dimension cannot generate novel descriptions. For instance, in [7], an intermediate meaning space based on the triplet of object, action, and scene is proposed to measure the similarity between image and sentence, where the top sentences are regarded as the generated sentences for the target image. Recently, a simple k-nearest neighbor retrieval model is utilized in [4] and the best or consensus caption is selected from the returned candidate captions, which even performs as well as several state-of-the-art language-based models.

Different from template-based and search-based models, language-based models aim to learn the probability distribution in the common space of visual content and textual sentence to generate novel sentences with more flexible syntactical structures. In this direction, recent works explore such probability distribution mainly by using neural networks for image captioning. For instance, in [31], Vinyals *et al.* propose an end-to-end neural networks architecture by utilizing LSTM to generate sentence for an image, which is further incorporated with attention mechanism in [33] to automatically focus on salient objects when generating corresponding words. More recently, in [32], high-level concepts/attributes are shown to obtain clear improvements on image captioning when injected into existing state-of-the-art RNN-based model and such attributes are further utilized as semantic attention [38] to enhance image captioning. In another work by Yao *et al.* [36], attribute/object detectors are developed and leveraged into image captioning to describe novel objects.

In short, our work in this paper belongs to the language-based models. Different from most of the aforementioned language-based models which mainly focus on sentence generation by solely depending on image representations [5, 16, 31, 33] or attributes [32], our work contributes by studying not only jointly exploiting image representations and attributes for image captioning, but also how the architecture can be better devised by exploring mutual relationship in between. It is also worth noting that [38] also involves attributes for image captioning. Ours is fundamentally different in the way that [38] is as a result of utilizing attributes to model semantic attention to the locally previous words, as opposed to holistically employing attributes as a kind of complementary representations in this work.

3. Boosting Image Captioning with Attributes

In this paper, we devise our CNN plus RNN architectures to generate descriptions for images under the umbrella of additionally incorporating the detected high-level attributes. Specifically, we begin this section by presenting the problem formulation. Then, an attributes prediction method by further exploring inter-attribute correlations is provided. Finally, five variants of our image captioning frameworks with attributes are investigated and discussed.

3.1. Problem Formulation

Suppose we have an image I to be described by a textual sentence $\mathcal{S}$, where $\mathcal{S} = \{w_1, w_2, ..., w_{N_s}\}$ consisting of N_s words. Let $\mathbf{I} \in \mathbb{R}^{D_v}$ and $\mathbf{w}_t \in \mathbb{R}^{D_s}$ denote the D_v-dimensional image representations of the image I and the D_s-dimensional textual features of the t-th word in sentence $\mathcal{S}$, respectively. Furthermore, we have feature vector $\mathbf{A} \in \mathbb{R}^{D_a}$ to represent the probability distribution over all the high-level attributes $\mathcal{A}$ for image I, where $\mathcal{A} = \{a_1, a_2, ..., a_{D_a}\}$ consisting of D_a attributes in the whole image captioning dataset. More details about how we mine and represent the attributes will be elaborated in Section 3.2. Taking inspiration from the recent successes of probabilistic sequence models leveraged in statistical machine translation [27] and image/video captioning [20, 21, 31], we aim to formulate our image captioning models in an end-to-end fashion based on RNNs which encode the given image and/or its detected attributes into a fixed dimensional vector and then decode it to the target output sentence. Hence, the sentence generation problem we explore here can be formulated by minimizing the following energy loss function as

$$E(\mathbf{I}, \mathbf{A}, \mathcal{S}) = -\log \Pr(\mathcal{S}|\mathbf{I}, \mathbf{A}), \tag{1}$$

which is the negative log probability of the correct textual sentence given the image and detected attributes.

Since the model produces one word in the sentence at each time step, it is natural to apply chain rule to model the joint probability over the sequential words. Thus, the $\log$ probability of the sentence is given by the sum of the $\log$ probabilities over the word and can be expressed as

$$\log \Pr(\mathcal{S}|\mathbf{I}, \mathbf{A}) = \sum_{t=1}^{N_s} \log \Pr(\mathbf{w}_t | \mathbf{I}, \mathbf{A}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1}). \tag{2}$$

By minimizing this loss, the contextual relationship among the words in the sentence can be guaranteed given the image and its detected attributes.

We formulate this task as a variable-length sequence to sequence problem and model the parametric distribution $\Pr(\mathbf{w}_t | \mathbf{I}, \mathbf{A}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1})$ in Eq.(2) with Long Short-Term Memory (LSTM) network, which is a widely used type of RNN and can capture long-term information in the sequential data by mapping sequences to sequences.

3.2. Attributes Prediction

An image generally contains not only multiple semantic attributes but also the interactions between the attributes. To detect attributes from images, one way is to train Fully Convolutional Networks (FCNs) by using the weakly-supervised multi-label classification approach of Multiple Instance Learning (MIL) in [6]. This method can easily predict the attributes probability distribution over massive attributes, but leaving the inherent semantic correlations between attributes unexploited as all the attributes detectors are learnt independently. To further explore the semantic correlations between attributes, a new MIL-based model with Inter-Attributes Correlations (MIL-IAC) is devised. Technically, for an attribute a_j, one image I is regarded as a positive bag of regions (instances) if a_j exists in image I's ground-truth sentences, and negative bag otherwise. By inputting all the bags into a noisy-OR MIL model [6], the probability of the bag b_I which contains attribute a_j is measured on the probabilities of all the regions in the bag:

$$\Pr_I^{a_j} = 1 - \prod_{r_i \in b_I} \left(1 - p_i^{a_j}\right), \tag{3}$$

where $p_i^{a_j}$ is the probability of the attribute a_j predicted by region r_i. We calculate $p_i^{a_j}$ through a sigmoid layer after the last convolutional layer in the fully convolutional network:

$$p_i^{a_j} = \frac{1}{1 + e^{-\mathbf{T}_j^\top \mathbf{r_i}}}, \tag{4}$$

where $\mathbf{T}_j \in \mathbb{R}^{D_t}$ denotes the detection parameter matrix in sigmoid layer for measuring the prediction score of j-th attribute a_j and $\mathbf{r_i}$ is the corresponding representation for image region r_i. In particular, the dimension of convolutional activations from the last convolutional layer is $x \times x \times D_t$ and D_t represents the representation dimension of each region, resulting in $x \times x$ response map which preserves the spatial dependency of the image. Then, a cross entropy loss is calculated based on the probabilities of all the attributes at the top of the whole FCNs architecture as

$$\begin{array}{l} l_c(I) = -\sum\limits_{j=1}^{D_a} \left[I_{(\mathbf{C}_j = 1)} \log \left(\Pr_I^{a_j} \right) \right. \\ \qquad \left. + \left(1 - I_{(\mathbf{C}_j = 1)}\right) \log \left(1 - \Pr_I^{a_j} \right) \right] \end{array}, \tag{5}$$

where $\Pr_I^{a_j}$ is measured as in Eq.(3), the indicator function $I_{condition} = 1$ if *condition* is true; otherwise $I_{condition} = 0$, and $\mathbf{C}_j$ denotes the j-th element in attributes label vector $\mathbf{C}$. Note that each element of the attributes label vector $\mathbf{C} \in \{0, 1\}^{D_a}$ is an attribute indicator. The indicator is 1 if the image contains this attribute otherwise the indicator is 0.

Inspired by the idea of structure preservation or manifold regularization in [19, 37], the inter-attribute correlation here is integrated into the learning of attributes detector as a regularizer in the sigmoid layer to further explore the inherent semantic correlations between attributes. This regularizer indicates that the detectors, i.e., detection parameter matrices in sigmoid layer, of semantically relevant attributes should be similar. The estimation of the underlying semantic correlations can be measured by the appropriate pairwise similarity between attributes. Specifically, the regularization of inter-attribute correlations could be given by

$$l_a(\mathbf{T}) = \sum_{m,n=1}^{D_a} \mathbf{S}_{mn} \|\mathbf{T}_m - \mathbf{T}_n\|^2, \tag{6}$$

where $\mathbf{S} \in \mathbb{R}^{D_a \times D_a}$ is the affinity matrix defined on the attributes, $\mathbf{T} \in \mathbb{R}^{D_t \times D_a}$ is the whole detection parameter matrix in sigmoid layer, and $\mathbf{T}_m$ denotes the m-th column of $\mathbf{T}$ representing the detection parameter matrix for attribute a_m. It is reasonable to minimize Eq.(6), since it will incur a heavy penalty if the distance between two similar detection parameter matrices is very far. There are many ways of defining the affinity matrix $\mathbf{S}$. Here, we calculate the elements through the normalized cosine similarity between two attributes:

$$\mathbf{S}_{mn} = \frac{\mathbf{a}_m \cdot \mathbf{a}_n}{\|\mathbf{a}_m\| \, \|\mathbf{a}_n\|}, \tag{7}$$

where $\mathbf{a}_m$ is a 300-dimensional word representation generated from word2vector neural network [17] for attribute a_m. Please note that each cosine similarity score $\mathbf{S}_{mn}$ is further linearly normalized into the range of $[0, 1]$.

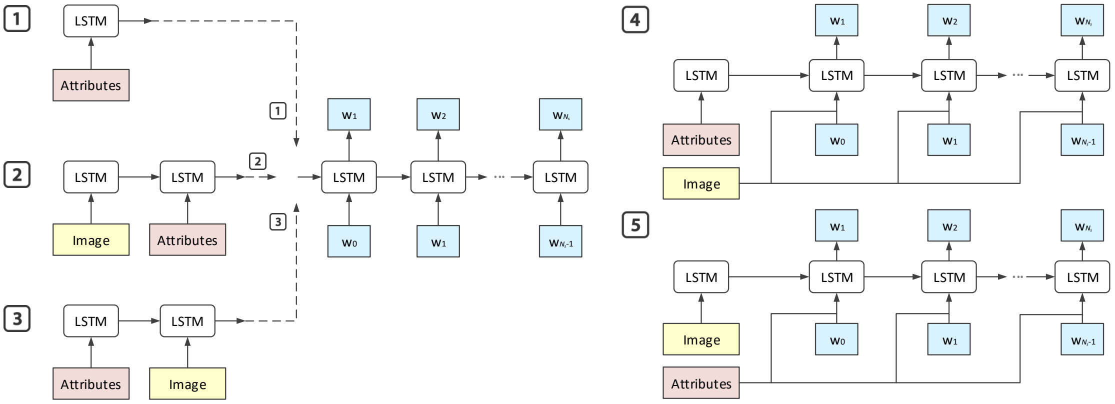


Figure 1. Five variants of our LSTM-A framework (better viewed in color).

By defining the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix with its elements defined as $\mathbf{D}_{mm} = \sum_n \mathbf{S}_{mn}$, Eq.(6) can be rewritten as

$$l_a(\mathbf{T}) = tr(\mathbf{TLT}^\top). \tag{8}$$

By minimizing this term, the inherent semantic corrections between attributes could be preserved in the learnt attributes detectors. The overall objective function integrates the cross entropy loss in Eq.(5) on the image set $\mathcal{I}$ and inter-attribute correlations regularization in Eq.(8). Hence, we obtain the following overall loss objective:

$$l = \lambda \sum_{I\in\mathcal{I}} l_c(I) + (1-\lambda)\, l_a(\mathbf{T}), \tag{9}$$

where λ is the tradeoff parameter. After optimizing the whole FCN architecture with the overall loss objective in Eq.(9), we complete the learning of our MIL-IAC attributes prediction model and treat its final prediction scores on all the attributes as $\mathbf{A}$.

3.3. Long Short-Term Memory with Attributes

Unlike the existing image captioning models in [5, 31] which solely encode image representations for sentence generation, our proposed Long Short-Term Memory with Attributes (LSTM-A) model additionally integrates the detected high-level attributes into LSTM. We devise five variants of LSTM-A for involvement of two design purposes. The first purpose is about where to feed attributes into LSTM and three architectures, i.e., LSTM-A$_1$ (leveraging only attributes), LSTM-A$_2$ (inserting image representations first) and LSTM-A$_3$ (feeding attributes first), are derived from this view. The second is about when to input attributes or image representations into LSTM and we design LSTM-A$_4$ (inputting image representations at each time step) and LSTM-A$_5$ (inputting attributes at each time step) for this purpose. An overview of LSTM-A is depicted in Figure 1.

3.3.1 LSTM-A$_1$ (Leveraging only Attributes)

Given the detected attributes, one natural way is to directly inject the attributes as representations at the initial time to inform the LSTM about the high-level attributes. This kind of architecture with only attributes input is named as LSTM-A$_1$. It is also worth noting that the attributes-based model in [32] is similar to LSTM-A$_1$ and can be regarded as one special case of our LSTM-A. Given the attribute representations $\mathbf{A}$ and the corresponding sentence $\mathbf{W} \equiv [\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_{N_s}]$, the LSTM updating procedure in LSTM-A$_1$ is as

$$\begin{array}{c} \mathbf{x}^{-1} = \mathbf{T}_a\mathbf{A}, \\ \mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t, \quad \text{and} \quad \mathbf{h}^t = f\left(\mathbf{x}^t\right), \; t \in \{0, \ldots, N_s - 1\}, \end{array} \tag{10}$$

where D_e is the dimensionality of LSTM input, $\mathbf{T}_a \in \mathbb{R}^{D_e \times D_a}$ and $\mathbf{T}_s \in \mathbb{R}^{D_e \times D_s}$ is the transformation matrix for attribute representations and textual features of word, respectively, f is the updating function within LSTM unit, and $\mathbf{h}^t$ is the cell output of the LSTM unit. Please note that for the input sentence $\mathbf{W} \equiv [\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_{N_s}]$, we take $\mathbf{w}_0$ as the start sign word to inform the beginning of sentence and $\mathbf{w}_{N_s}$ as the end sign word which indicates the end of sentence. Both of the special sign words are included in our vocabulary. Most specifically, at the initial time step, the attribute representations are transformed as the input to LSTM, and then in the next steps, word embedding $\mathbf{x}^t$ will be input into the LSTM along with the previous step's hidden state $\mathbf{h}^{t-1}$. In each time step (except the initial step), we use the LSTM cell output $\mathbf{h}^t$ to predict the next word through a softmax layer.

3.3.2 LSTM-A$_2$ (Inserting image first)

To further leverage both image representations and high-level attributes in the encoding stage of our LSTM-A, we design the second architecture LSTM-A$_2$ by treating both of them as atoms in the input sequence to LSTM. Specifically, at the initial step, the image representations $\mathbf{I}$ are firstly transformed into LSTM to inform the LSTM about the image content, followed by the attribute representations $\mathbf{A}$ which are encoded into LSTM at the next time step to inform the high-level attributes. Then, LSTM decodes each output word based on previous word $\mathbf{x}^t$ and previous step's hidden state $\mathbf{h}^{t-1}$, which is similar to the decoding stage in LSTM-A$_1$. The LSTM updating procedure in LSTM-A$_2$ is designed as

$$\begin{gathered}\mathbf{x}^{-2} = \mathbf{T}_v\mathbf{I} \text{ and } \mathbf{x}^{-1} = \mathbf{T}_a\mathbf{A}, \\ \mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t, \quad \text{and} \quad \mathbf{h}^t = f\left(\mathbf{x}^t\right), \ t \in \{0, \ldots, N_s - 1\}, \end{gathered} \tag{11}$$

where $\mathbf{T}_v \in \mathbb{R}^{D_e \times D_v}$ is the transformation matrix for image representations.

3.3.3 LSTM-A$_3$ (Feeding attributes first)

The third design LSTM-A$_3$ is similar to LSTM-A$_2$ as both designs utilize image representations and high-level attributes to form the input sequence to LSTM in the encoding stage, except that the orders of encoding are different. In LSTM-A$_3$, the attribute representations are firstly encoded into LSTM and then the image representations are transformed into LSTM at the second time step. The whole LSTM updating procedure in LSTM-A$_3$ is as

$$\begin{gathered}\mathbf{x}^{-2} = \mathbf{T}_a\mathbf{A} \text{ and } \mathbf{x}^{-1} = \mathbf{T}_v\mathbf{I}, \\ \mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t, \quad \text{and} \quad \mathbf{h}^t = f\left(\mathbf{x}^t\right), \ t \in \{0, \ldots, N_s - 1\}. \end{gathered} \tag{12}$$

3.3.4 LSTM-A$_4$ (Inputting image each time step)

Different from the former three designed architectures which mainly inject high-level attributes and image representations at the encoding stage of LSTM, we next modify the decoding stage in our LSTM-A by additionally incorporating image representations or high-level attributes. More specifically, in LSTM-A$_4$, the attribute representations are injected once at the initial step to inform the LSTM about the high-level attributes, and then image representations are fed at each time step as an extra input to LSTM to emphasize the image content frequently among memory cells in LSTM. Hence, the LSTM updating procedure in LSTM-A$_4$ is:

$$\begin{gathered}\mathbf{x}^{-1} = \mathbf{T}_a\mathbf{A}, \\ \mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t + \mathbf{T}_v\mathbf{I}, \text{ and } \mathbf{h}^t = f\left(\mathbf{x}^t\right), \ t \in \{0, \ldots, N_s - 1\}. \end{gathered} \tag{13}$$

3.3.5 LSTM-A$_5$ (Inputting attributes each time step)

The last design LSTM-A$_5$ is similar to LSTM-A$_4$ except that it firstly encodes image representations and then feeds attribute representations as an additional input to LSTM at each step in decoding stage to emphasize the high-level attributes frequently. Accordingly, the LSTM updating procedure in LSTM-A$_5$ is as

$$\begin{gathered}\mathbf{x}^{-1} = \mathbf{T}_v\mathbf{I}, \\ \mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t + \mathbf{T}_a\mathbf{A}, \text{and } \mathbf{h}^t = f\left(\mathbf{x}^t\right), \ t \in \{0, \ldots, N_s - 1\}. \end{gathered} \tag{14}$$

4. Experiments

We conducted the experiments and evaluated our approaches on COCO captioning dataset (COCO) [13].

4.1. Dataset and Experimental Settings

The **dataset**, COCO, is the most popular benchmark for image captioning, which contains 82,783 training images and 40,504 validation images. There are 5 human-annotated descriptions per image. As the annotations of the official testing set are not publicly available, we follow the widely used settings in [38, 39] and take 82,783 images for training, 5,000 for validation and 5,000 for testing.

Data Preprocessing. Following [10], we convert all the descriptions in training set to lower case and discard rare words which occur less than 5 times, resulting in the final vocabulary with 8,791 unique words in COCO dataset.

Features and Parameter Settings. Each word in the sentence is represented as "one-hot" vector (binary index vector in a vocabulary). For image representations, we take the output of 1,024-way $pool5/7 \times 7\_s1$ layer from GoogleNet [28] pre-trained on Imagenet ILSVRC12 dataset [25]. For attribute representations, we select 1,000 most common words on COCO as the high-level attributes and train our MIL-IAC attributes prediction model purely on the training data of COCO, resulting in the final 1,000-way vector of probabilities of attributes. The tradeoff parameter λ is empirically set as 0.8. The dimension of the input and hidden layers in LSTM of LSTM-A are both set to 1,024.

Implementation Details. We mainly implement our image captioning models based on Caffe [9], which is one of widely adopted deep learning frameworks. Specifically, with an initial learning rate 0.01 and mini-batch size of 1,024, the objective value can decrease to 25% of the initial loss and reach a reasonable result after 50,000 iterations.

Testing Strategies. For sentence generation in testing stage, we adopt the beam search strategy which selects the top-k best sentences at each time step and considers them as the candidates to generate new top-k best sentences at the next time step. The beam size k is empirically set to 3.

Evaluation Metrics. For the evaluation of our proposed models, we adopt five types of metrics: BLEU@N [22], METEOR [2], ROUGE-L [12], CIDEr-D [30] and SPICE [1]. All the metrics are computed by using the codes[1] released by COCO Evaluation Server [3].

4.2. Compared Approaches

To verify the merit of our LSTM-A models, we compared the following state-of-the-art methods.

(1) NIC & LSTM [31]: NIC is the standard RNN-based model which only injects image into LSTM at the initial time step. We directly extract results reported in [38] and

[1] https://github.com/tylin/coco-caption

Table 1. Performance of our proposed models and other state-of-the-art methods on COCO, where B@N, M, R, C and S are short for BLEU@N, METEOR, ROUGE-L, CIDEr-D and SPICE scores. All values are reported as percentage (%).

Model	**B@1**	**B@2**	**B@3**	**B@4**	**M**	**R**	**C**	**S**
NIC [31]	66.6	45.1	30.4	20.3	-	-	-	-
LRCN [5]	69.7	51.9	38.0	27.8	22.9	50.8	83.7	15.8
HA [33]	71.8	50.4	35.7	25	23	-	-	-
SA [33]	70.7	49.2	34.4	24.3	23.9	-	-	-
ATT [38]	70.9	53.7	40.2	30.4	24.3	-	-	-
SC [39]	72	54.6	40.4	29.8	24.5	-	95.9	-
LSTM [31]	68.4	51.2	38	28.4	23.1	50.7	84.3	16
LSTM-A$_1$	72.9	56.2	42.4	31.9	25.1	53.4	97.5	18.1
LSTM-A$_2$	73.3	56.5	42.7	32.2	25.3	53.9	99.1	18.3
LSTM-A$_3$	**73.5**	56.6	42.9	32.4	**25.5**	53.9	99.8	18.5
LSTM-A$_4$	72.1	55.5	41.7	31.4	24.9	53.2	95.7	17.8
LSTM-A$_5$	73.4	**56.7**	**43.0**	**32.6**	25.4	**54.0**	**100.2**	**18.6**
LSTM-A*	**95.7**	**82.5**	**68.5**	**55.9**	**34.1**	**67.3**	**150.5**	**26.8**

name this run as NIC. Moreover, for fair comparison, we also include our implementation of NIC, named as LSTM.

(2) LRCN [5]: LRCN inputs both image representations and previous word into LSTM at each time step.

(3) Hard-Attention (HA) & Soft-Attention (SA) [33]: Spatial attention on convolutional features of an image is incorporated into the encoder-decoder framework through two kinds of mechanisms: 1) "hard" stochastic attention equivalently by reinforce learning and 2) "soft" deterministic attention with standard back-propagation.

(4) ATT [38]: ATT utilizes attributes as semantic attention to combine image and attributes in RNN for captioning.

(5) Sentence-Condition (SC) [39]: Sentence-condition exploits text-conditional semantic attention to generate semantic guidance for sentence generation by conditioning image features on current text content.

(6) MSR Captivator [4]: MSR Captivator employs both Multimodal Recurrent Neural Network (MRNN) and Maximum Entropy Language Model (MELM) [6] for sentence generation. Deep Multimodal Similarity Model (DMSM) [6] is further exploited for sentence re-ranking.

(7) CaptionBot [29]: CaptionBot is a publicly image captioning system[2] which is mainly built on vision models by using Deep residual networks (ResNets) [8] to detect visual concepts, MELM [6] language model for sentence generation and DMSM [6] for caption ranking.

(8) LSTM-A: LSTM-A$_1$ $\sim$ LSTM-A$_5$ are five variants derived from our proposed LSTM-A framework. In addition, LSTM-A* is an oracle run that inputs ground-truth attributes into the LSTM-A$_3$ architecture.

4.3. Performance Comparison on COCO

Table 1 shows the performances of different models on COCO image captioning dataset. It is worth noting that the performances of different approaches here are based on different image representations. Specifically, VGG architecture [26] is utilized as image feature extractor in the methods of Hard-Attention & Soft-Attention and Sentence-Condition, while GoogleNet [28] is exploited in NIC, LRCN, ATT, LSTM and our LSTM-A. In view that the GoogleNet and VGG features are comparable, we compare directly with results. Overall, the results across eight evaluation metrics consistently indicate that our proposed LSTM-A exhibits better performance than all the state-of-the-art techniques including non-attention models (NIC, LSTM, LRCN) and attention-based methods (Hard-Attention, Soft-Attention, ATT, Sentence-Condition). In particular, the CIDEr-D and SPICE can achieve 100.2% and 18.6%, respectively, when extracting image representations by GoogleNet. LSTM-A$_1$ inputting only high-level attributes as representations makes the relative improvement over LSTM which feeds into image representations instead by 12.3%, 8.7%, 5.3%, 15.7% and 13.1% in BLEU@4, METEOR, ROUGR-L, CIDEr-D and SPICE, respectively. The results basically indicate the advantage of exploiting high-level attributes than image representations for image captioning. Furthermore, by additionally incorporating attributes to LSTM model, LSTM-A$_2$, LSTM-A$_3$ and LSTM-A$_5$ lead to a performance boost, indicating that image representations and attributes are complementary and thus have mutual reinforcement for image captioning. Similar in spirit, LSTM-A$_4$ improves LRCN by further taking attributes into account. There is a significant performance gap between ATT and LSTM-A$_5$. Though both runs involve the utilization of image representations and attributes, they are fundamentally different in the way that the performance of ATT is as a result of modulating the strength of attention on attributes to the previous words, and LSTM-A$_5$ is by employing attributes as auxiliary knowledge to complement image representations. This somewhat reveals the weakness of semantic attention model, where the prediction errors will accumulate along the generated sequence.

[2] `https://www.captionbot.ai`

Table 2. Leaderboard of the published state-of-the-art image captioning models on the online COCO testing server, where B@N, M, R, and C are short for BLEU@N, METEOR, ROUGE-L, and CIDEr-D scores. All values are reported as percentage (%).

Model	B@1		B@2		B@3		B@4		M		R		C	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
LSTM-A$_3$ (Ours)	**78.7**	**93.7**	**62.7**	**86.7**	**47.6**	**76.5**	**35.6**	**65.2**	**27**	**35.4**	**56.4**	**70.5**	**116**	**118**
Watson Multimodal [24]	77.3	92.9	60.9	85.6	46.1	75.1	34.4	63.6	26.8	35.3	55.9	70.4	112.3	114.6
G-RMI(PG-SPIDEr-TAG) [14]	75.1	91.6	59.1	84.2	44.5	73.8	33.1	62.4	25.5	33.9	55.1	69.4	104.2	107.1
MetaMind/VT_GT [15]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9
reviewnet [35]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	53.3	68.6	96.5	96.9
ATT [38]	73.1	90	56.5	81.5	42.4	70.9	31.6	59.9	25	33.5	53.5	68.2	94.3	95.8
Google [31]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	53	68.2	94.3	94.6
MSR Captivator [4]	71.5	90.7	54.3	81.9	40.7	71	30.8	60.1	24.8	33.9	52.6	68	93.1	93.7

Table 3. BLEU@4, METEOR, ROUGE-L, CIDEr-D, and SPICE scores of our proposed LSTM-A$_3$ with attributes learnt by different attributes prediction models on COCO.

Model	B@4	M	R	C	S
Fine-tune	30.2	24.3	52.4	91.7	17.2
MIL [6]	32.1	25.2	53.7	98.4	18.2
MIL-IAC	**32.4**	**25.5**	**53.9**	**99.8**	**18.5**

Compared to LSTM-A$_1$, LSTM-A$_2$ which is augmented by integrating image representations performs better, but the performances are lower than LSTM-A$_3$. The results indicate that LSTM-A$_3$, in comparison, benefits from the mechanism of first feeding high-level attributes into LSTM instead of starting from inserting image representations in LSTM-A$_2$. The chance that a good start point can be attained and lead to performance gain is better. LSTM-A$_4$ feeding the image representations at each time step yields inferior performances to LSTM-A$_3$, which only inputs image representations once. We speculate that this may because the noise in the image can be explicitly accumulated and thus the network overfits more easily. In contrast, the performances of LSTM-A$_5$ which feeds attributes at each time step show the improvements on LSTM-A$_3$. The results demonstrate that the high-level attributes are more accurate and easily translated into human understandable sentence. Among the five proposed LSTM-A architectures, LSTM-A$_3$ achieves the best performances in terms of BLEU@1 and METEOR, while LSTM-A$_5$ performs the best in other six evaluation metrics. The performances of the oracle run LSTM-A* could be regarded as the upper bound of employing attributes in our framework and lead to large performance gain against LSTM-A$_3$. Such an upper bound enables us to obtain more insights on the factor accounting for the success of the current attribute augmented architecture and also provides guidance to future research in this direction. More specifically, the results, on one hand, indicate the advantage and great potential of leveraging attributes for boosting image captioning, and on the other, suggest that more efforts are further required towards mining and representing attributes more effectively.

4.4. Evaluation of Attributes Prediction Model

We further verify the effectiveness of our MIL-IAC attributes prediction model. We compared two baselines here. One is to directly fine-tune the VGG architecture with cross entropy loss for attributes prediction, named as Fine-tune, and the other, namely MIL, exploits a weakly supervised MIL model [6] based on VGG to learn region-based detectors for attributes. Table 3 compares the sentence generation performances of our LSTM-A$_3$ model with attributes learnt by different attributes prediction models on COCO dataset. Compared to Fine-tune, MIL method using region-based detectors consistently exhibits better performance across different evaluation metric. Moreover, by additionally exploring the inter-attribute correlations in MIL framework, our proposed MIL-IAC leads to larger performance gains.

4.5. Performance on COCO Online Testing Server

We also submitted our best run in terms of METEOR, i.e., LSTM-A$_3$, to online COCO testing server and evaluated the performance on official testing set. Table 2 shows the performance Leaderboard on official testing image set with 5 reference captions (c5) and 40 reference captions (c40). Please note that here we utilize the outputs of 2,048-way $pool5$ layer from ResNet-152 as image representations and train the attribute detectors by ResNet-152 in our final submission. Moreover, inspired by [24], we adopt the policy gradient optimization to specifically boost CIDEr-D performance. The latest top-8 performing methods which have been officially published are included in the table. Compared to the top performing methods on the leaderboard, our proposed LSTM-A$_3$ achieves the best performances across all the evaluation metrics on both c5 and c40 testing sets.

4.6. Human Evaluation

To better understand how satisfactory are the sentences generated from different methods, we also conducted a human study to compare our LSTM-A$_3$ against three approaches, i.e., CaptionBot, LRCN and LSTM. A total number of 12 evaluators (6 females and 6 males) from different education backgrounds, including computer science (4), business (2), linguistics (2) and engineering (4), are invited and a subset of 1K images is randomly selected from testing set for the subjective evaluation. The evaluation process is as follows. All the evaluators are organized into two groups. We show the first group all the four sentences generated by each approach plus five human-annotated sentences and ask

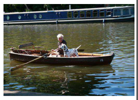
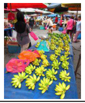
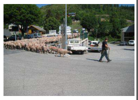
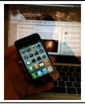
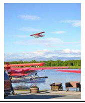
	**Attributes:** boat: 1 water: 0.838 man: 0.762 riding: 0.728 dog: 0.547 small: 0.485 person: 0.471 river: 0.461	**Generated Sentences:** **LSTM:** a group of people on a boat in the water **CaptionBot:** I think it's a man with a small boat in a body of water. **LSTM-A$_3$:** a man and a dog on a boat in the water	**Ground Truth:** ① an image of a man in a boat with a dog ② a person on a rowboat with a dalmatian dog on the boat ③ old woman rowing a boat with a dog
	**Attributes:** bananas: 1 people: 0.956 market: 0.708 standing: 0.612 outdoor: 0.558 blue: 0.514 large: 0.407 table: 0.381	**Generated Sentences:** **LSTM:** a group of people standing around a market **CaptionBot:** I think it's a bunch of yellow flowers. **LSTM-A$_3$:** a group of people standing around a bunch of bananas	**Ground Truth:** ① bunches of bananas for sale at an outdoor market ② a person at a table filled with bananas ③ there are many bananas layer across this table at a farmers market
	**Attributes:** sheep: 0.976 herd: 0.778 street: 0.702 walking: 0.702 road: 0.635 man: 0.555 standing: 0.430 animals: 0.388	**Generated Sentences:** **LSTM:** a man riding a skateboard down a street **CaptionBot:** I think it's a group of people walking down the road. **LSTM-A$_3$:** a man walking down a street with a herd of sheep	**Ground Truth:** ① a man walks while a large number of sheep follow ② a man leading a herd of sheep down the sheep ③ the man is walking a herd of sheep on the road through a town
	**Attributes:** phone: 0.867 cell: 0.839 computer: 0.735 laptop: 0.641 keyboard: 0.581 screen: 0.546 holding: 0.505 person: 0.334	**Generated Sentences:** **LSTM:** a cell phone sitting on top of a table **CaptionBot:** I think it's a laptop that is on the phone. **LSTM-A$_3$:** a person holding a cell phone in front of a laptop	**Ground Truth:** ① a smart phone being held up in front of a lap top ② the person is holding his cell phone while on his laptop ③ someone holding a cell phone in front of a laptop
	**Attributes:** flying: 0.997 airplane: 0.957 plane: 0.941 water: 0.893 red: 0.837 lake: 0.751 white: 0.566 sky: 0.565	**Generated Sentences:** **LSTM:** a group of people flying kites in the sky **CaptionBot:** I think it's a plane is flying over the water. **LSTM-A$_3$:** a red and white plane flying over a body of water	**Ground Truth:** ① a plane with water skies for landing gear coming in for a landing at a lake ② a plane flying through a sky above a lake ③ a red and white plane is flying over some water

Figure 2. Attributes and sentences generation results on COCO. The attributes are detected by our attributes prediction model and the output sentences are generated by 1) LSTM, 2) CaptionBot[2], 3) our LSTM-A$_3$ and 4) Ground Truth: three ground truth sentences.

Table 4. User study on two criteria: M1 - percentage of captions generated by different methods that are evaluated as better/equal to human caption; M2 - percentage of captions that pass Turing Test.

	Human	LSTM-A$_3$	CaptionBot	LSTM	LRCN
M1	-	**64.9**	58.2	49.2	43.9
M2	**90.1**	**75.3**	66.3	57.3	55.9

them the question: Do the systems produce captions resembling human-generated sentences? In contrast, we show the second group once only one sentence generated by different approach or human annotation and they are asked: Can you determine whether the given sentence has been generated by a system or by a human being? From evaluators' responses, we calculate two metrics: 1) M1: percentage of captions that are evaluated as better or equal to human caption; 2) M2: percentage of captions that pass the Turing Test. Table 4 lists the result of the user study. Overall, our LSTM-A$_3$ is clearly the winner for all two criteria. In particular, the percentage achieves 64.9% and 75.3% in terms of M1 and M2, respectively, making the absolute improvement over the best competitor CaptionBot by 6.7% and 9%.

4.7. Qualitative Analysis

Figure 2 showcases a few sentence examples generated by different methods, the detected high-level attributes, and human-annotated ground truth sentences. From these exemplar results, it is easy to see that all of these automatic methods can generate somewhat relevant sentences, while our proposed LSTM-A$_3$ can predict more relevant keywords by jointly exploiting high-level attributes and image representations for image captioning. For example, compared to subject term "a group of people" and "a man" in the sentence generated by LSTM and CaptionBot respectively, "a man and a dog" in our LSTM-A$_3$ is more precise to describe the image content in the first image, since the keyword "dog" is one of the detected attributes and directly injected into LSTM to guide the sentence generation. Similarly, verb term "holding" which is also detected as one high-level attribute presents the fourth image more exactly. Moreover, our LSTM-A$_3$ generate more descriptive sentence by enriching the semantics with high-level attributes. For instance, with the detected adjective "red," the generated sentence "a red and white plane flying over a body of water" of the fifth image depicts the image content more comprehensive. We refer the readers to supplementary materials for more examples.

5. Discussions and Conclusions

We have presented Long Short-Term Memory with Attributes (LSTM-A) architectures which explores both image representations and high-level attributes for image captioning. Particularly, we detect attributes by additionally exploring the inter-attribute correlations in the Multiple Instance Learning framework and study the problem of augmenting high-level attributes from images to complement image representations for enhancing sentence generation. To verify our claim, we have devised variants of architectures by modifying the placement and moment, where and when to feed into the two kinds of representations. Experiments conducted on COCO image captioning dataset validate our proposal and analysis. Performance improvements are observed when comparing to other captioning techniques.

Our future works are as follows. First, more attributes will be learnt from large-scale image benchmarks, e.g., YFCC-100M dataset, and integrated into image captioning. Second, how to generate free-form and open-vocabulary sentences with the learnt attributes is also expected.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

[2] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*, 2005.

[3] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[4] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. Language models for image captioning: The quirks and what works. In *ACL*, 2015.

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[6] H. Fang, S. Gupta, et al. From captions to visual concepts and back. In *CVPR*, 2015.

[7] 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.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *MM*, 2014.

[10] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[11] G. Kulkarni, V. Premraj, et al. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. on PAMI*, 2013.

[12] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*, 2004.

[13] 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. In *ECCV*, 2014.

[14] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Optimization of image description metrics using policy gradient methods. *arXiv preprint arXiv:1612.00370*, 2016.

[15] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

[16] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. In *NIPS Workshop on Deep Learning*, 2014.

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

[18] M. Mitchell, X. Han, et al. Midge: Generating image descriptions from computer vision detections. In *EACL*, 2012.

[19] Y. Pan, Y. Li, T. Yao, T. Mei, H. Li, and Y. Rui. Learning deep intrinsic video representation by exploring temporal coherence and graph structure. In *IJCAI*, 2016.

[20] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[21] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *CVPR*, 2017.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[23] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[24] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[25] 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. *IJCV*, 2015.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

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

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[29] K. Tran, X. He, L. Zhang, J. Sun, C. Carapcea, C. Thrasher, C. Buehler, and C. Sienkiewicz. Rich image captioning in the wild. *arXiv preprint arXiv:1603.09016*, 2016.

[30] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[31] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[32] Q. Wu, C. Shen, L. Liu, A. Dick, and A. v. d. Hengel. What value do explicit high level concepts have in vision to language problems? In *CVPR*, 2016.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[34] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[35] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov. Review networks for caption generation. In *NIPS*, 2016.

[36] T. Yao, Y. Pan, Y. Li, and T. Mei. Incorporating copying mechanism in image captioning for learning novel objects. In *CVPR*, 2017.

[37] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015.

[38] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[39] L. Zhou, C. Xu, P. Koch, and J. J. Corso. Image caption generation with text-conditional semantic attention. *arXiv preprint arXiv:1606.04621*, 2016.