Boosting Image Captioning with Attributes

Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, Tao Mei
Microsoft Research Asia
`{tiyao, v-yipan, v-yehl, v-zhqiu, tmei}@microsoft.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. To incorporate attributes, 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 achieves superior results when compared to state-of-the-art deep models. Most remarkably, we obtain METEOR/CIDEr-D of 25.2%/98.6% on testing data of widely used and publicly available splits in (Karpathy & Fei-Fei, 2015) when extracting image representations by GoogleNet and achieve to date top-1 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 (Szegedy et al., 2015; Simonyan & Zisserman, 2015; He et al., 2016). 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 or 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 (Donahue et al., 2015; Vinyals et al., 2015; Xu et al., 2015; You et al., 2016) are from the advances by using RNNs in machine translation (Sutskever et al., 2014), 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. Furthermore, attributes are properties observed in images with rich semantic cues and have been proved to be effective in visual recognition (Parikh & Grauman, 2011). 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. More importantly, 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 place-

ments 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.

2 RELATED WORK

The research on image captioning has proceeded along three different dimensions: template-based methods (Kulkarni et al., 2013; Yang et al., 2011; Mitchell et al., 2012), search-based approaches (Farhadi et al., 2010; Ordonez et al., 2011; Devlin et al., 2015), and language-based models (Donahue et al., 2015; Kiros et al., 2014; Mao et al., 2014; Vinyals et al., 2015; Xu et al., 2015; Wu et al., 2016; You et al., 2016).

The first direction, template-based methods, predefine the template for sentence generation which follows some specific rules of language grammar and split sentence into several parts (e.g., subject, verb, and object). With such sentence fragments, many works align each part with image content and then generate the sentence for the image. Obviously, most of them highly depend on the templates of sentence and always generate sentence 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 (Kulkarni et al., 2013). Similar in spirit, Yang *et al.* utilize Hidden Markov Model (HMM) to select the best objects, scenes, verbs, and prepositions with the highest log-likelihood ratio for template-based sentence generation in (Yang et al., 2011). Furthermore, the traditional simple template is extended to syntactic trees in (Mitchell et al., 2012) which also starts from detecting attributes from image as description anchors and then connecting ordered objects with a syntactically well-formed tree, followed by adding necessary descriptive information.

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 (Farhadi et al., 2010), 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. Ordonez *et al.* (Ordonez et al., 2011) search images in a large captioned photo collection by using the combination of object, stuff, people, and scene information and transfer the associated sentences to the query image. Recently, a simple k-nearest neighbor retrieval model is utilized in (Devlin et al., 2015) 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 using neural networks for image captioning. Kiros *et al.* (Kiros et al., 2014) take the neural networks to generate sentence for an image by proposing a multimodal log-bilinear neural language model. In (Vinyals et al., 2015), 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 (Xu et al., 2015) to automatically focus on salient objects when generating corresponding words. More recently, in (Wu et al., 2016), high-level concepts/attributes are shown to obtain clear improvements on image captioning task when injected into existing state-of-the-art RNN-based model and such visual attributes are further utilized as semantic attention in (You et al., 2016) to enhance image captioning.

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 (Donahue et al., 2015; Kiros et al., 2014; Mao et al., 2014; Vinyals et al., 2015; Xu et al., 2015) or high-level attributes (Wu et al., 2016), 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 (You et al., 2016) also additionally involve attributes for image captioning. Ours is fundamentally different in the way that (You et al., 2016) 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 and followed by five variants of our image captioning frameworks with attributes.

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 the high-level attributes for image I. Specifically, we train the attribute detectors by using the weakly-supervised approach of Multiple Instance Learning (MIL) in (Fang et al., 2015) and treat the final image-level response probabilities of all the attributes as $\mathbf{A}$.

Inspired by the recent successes of probabilistic sequence models leveraged in statistical machine translation (Bahdanau et al., 2015; Sutskever et al., 2014), 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 representations 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

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

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 $\Pr(\mathbf{w}_t|\mathbf{I}, \mathbf{A}, \mathbf{w}_0, \dots, \mathbf{w}_{t-1})$ in Eq.(2) with Long Short-Term Memory (LSTM), which is a widely used type of RNN. The vector formulas for a LSTM layer forward pass are summarized as below. For time step t, $\mathbf{x}^t$ and $\mathbf{h}^t$ are the input and output vector respectively, $\mathbf{T}$ are input weights matrices, $\mathbf{R}$ are recurrent weight matrices and $\mathbf{b}$ are bias vectors. Sigmoid σ and hyperbolic tangent ϕ are element-wise non-linear activation functions. The dot product of two vectors is denoted with $\odot$. Given inputs $\mathbf{x}^t$, $\mathbf{h}^{t-1}$ and $\mathbf{c}^{t-1}$, the LSTM unit updates for time step t are:

$$\mathbf{g}^t = \phi(\mathbf{T}_g \mathbf{x}^t + \mathbf{R}_g \mathbf{h}^{t-1} + \mathbf{b}_g), \quad \mathbf{i}^t = \sigma(\mathbf{T}_i \mathbf{x}^t + \mathbf{R}_i \mathbf{h}^{t-1} + \mathbf{b}_i),$$
$$\mathbf{f}^t = \sigma(\mathbf{T}_f \mathbf{x}^t + \mathbf{R}_f \mathbf{h}^{t-1} + \mathbf{b}_f), \quad \mathbf{c}^t = \mathbf{g}^t \odot \mathbf{i}^t + \mathbf{c}^{t-1} \odot \mathbf{f}^t,$$
$$\mathbf{o}^t = \sigma(\mathbf{T}_o \mathbf{x}^t + \mathbf{R}_o \mathbf{h}^{t-1} + \mathbf{b}_o), \quad \mathbf{h}^t = \phi(\mathbf{c}^t) \odot \mathbf{o}^t,$$

where $\mathbf{g}^t$, $\mathbf{i}^t$, $\mathbf{f}^t$, $\mathbf{c}^t$, $\mathbf{o}^t$, and $\mathbf{h}^t$ are cell input, input gate, forget gate, cell state, output gate, and cell output of the LSTM, respectively.

3.2 LONG SHORT-TERM MEMORY WITH ATTRIBUTES

Unlike the existing image captioning models in (Donahue et al., 2015; Vinyals et al., 2015) 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
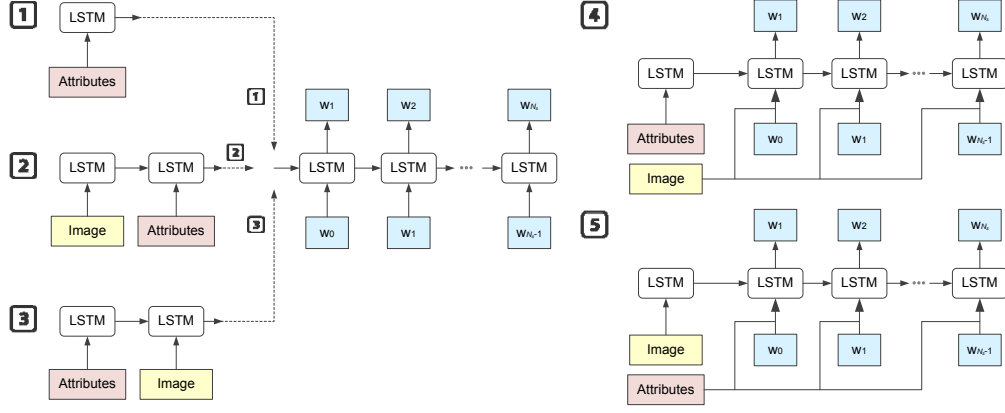
Figure 1: Five variants of our LSTM-A framework (better viewed in color).

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 architectures is depicted in Figure 1.

3.2.1 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 (Wu et al., 2016) 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

$$\mathbf{x}^{-1} = \mathbf{T}_a \mathbf{A},$$
$$\mathbf{x}^t = \mathbf{T}_s \mathbf{w}_t, \;\; t \in \{0, \dots, N_s - 1\} \quad \text{and} \quad \mathbf{h}^t = f\left(\mathbf{x}^t\right), \;\; t \in \{0, \dots, N_s - 1\},$$

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, and f is the updating function within LSTM unit. Please note that for the input sentence $\mathbf{W} \equiv [\mathbf{w}_0, \dots, \mathbf{w}_{N_s}]$, 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. Here a softmax layer is applied after the LSTM layer to produce a probability distribution over all the D_s words in the vocabulary as

$$\text{Pr}_{t+1}\left(w_{t+1}\right) = \frac{\exp\left\{\mathbf{T}_h^{(w_{t+1})}\mathbf{h}^t\right\}}{\sum\limits_{w \in \mathcal{W}} \exp\left\{\mathbf{T}_h^{(w)}\mathbf{h}^t\right\}}, \tag{3}$$

where $\mathcal{W}$ is the word vocabulary space and $\mathbf{T}_h^{(w)}$ is the parameter matrix in softmax layer.

3.2.2 LSTM-A_2 (INSERTING IMAGE REPRESENTATIONS 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

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

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

3.2.3 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

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

3.2.4 LSTM-A$_4$ (INPUTTING IMAGE REPRESENTATIONS AT 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:

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

3.2.5 LSTM-A$_5$ (INPUTTING ATTRIBUTES AT 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

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

4 EXPERIMENTS

We conducted our experiments on COCO captioning dataset (COCO) (Lin et al., 2014) and evaluated our approaches for image captioning.

4.1 DATASET

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 prior works (You et al., 2016; Zhou et al., 2016) and take 82,783 images for training, 5,000 for validation and 5,000 for testing.

4.2 EXPERIMENTAL SETTINGS

Data Preprocessing. Following (Karpathy & Fei-Fei, 2015), 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 (Szegedy et al., 2015) pre-trained on Imagenet ILSVRC12 dataset (Russakovsky et al., 2015). For attribute representations, we select 1,000 most common words on COCO as the high-level attributes and train the attribute detectors with MIL model (Fang et al., 2015) purely on the training data of COCO, resulting in the final 1,000-way vector of probabilities of attributes. The dimensionality of the input and hidden layers in LSTM are both set to 1,024.

Implementation Details. We mainly implement our image captioning models based on Caffe (Jia et al., 2014), which is one of widely adopted deep learning frameworks. Specifically, with an initial learning rate 0.01 and mini-batch size set 1,024, the objective value can decrease to 25% of the initial loss and reach a reasonable result after 50,000 iterations (about 123 epochs).

Testing Strategies. For sentence generation in testing stage, there are two common strategies. One is to choose the word with maximum probability at each time step and set it as LSTM input for next time step until the end sign word is emitted or the maximum length of sentence is reached. The other strategy is beam search 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. We adopt the second strategy and the beam size k is empirically set to 3.

Moreover, to avoid model-level overfitting, we utilize ensembling strategy to fuse the prediction results of 5 identical models as previous works (Vinyals et al., 2015; You et al., 2016). Please note that all the 5 identical models are trained with different initializations separately.

Evaluation Metrics. For the evaluation of our proposed models, we adopt four metrics: BLEU@N (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004), and CIDEr-D (Vedantam et al., 2015). All the metrics are computed by using the codes[1] released by COCO Evaluation Server (Chen et al., 2015).

4.3 COMPARED APPROACHES

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

- NIC & LSTM (Vinyals et al., 2015): NIC attempts to directly translate from image pixels to natural language with a single deep neural network. The image representations are only injected into LSTM at the initial time step. We directly extract the results reported in (You et al., 2016) and name this run as NIC. Furthermore, for fair comparison, we also include one run LSTM which is our implementation of NIC.

- LRCN (Donahue et al., 2015): LRCN inputs both image representations and previous word into LSTM at each time step for sentence generation.

- Hard-Attention & Soft-Attention (Xu et al., 2015): Spatial attention on convolutional features of an image is incorporated into the encoder-decoder framework through two kinds of mechanisms: 1) "hard" stochastic attention mechanism equivalently by reinforce learning (Hard-Attention) and 2) "soft" deterministic attention mechanism with standard back-propagation (Soft-Attention).

- ATT (You et al., 2016): ATT utilizes attributes as semantic attention to combine image representations and attributes in RNN for image captioning.

---

[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, 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
NIC (Vinyals et al., 2015)	66.6	45.1	30.4	20.3	-	-	-
LRCN (Donahue et al., 2015)	62.8	44.2	30.4	21	-	-	-
Hard-Attention (Xu et al., 2015)	71.8	50.4	35.7	25	23	-	-
Soft-Attention (Xu et al., 2015)	70.7	49.2	34.4	24.3	23.9	-	-
ATT (You et al., 2016)	70.9	53.7	40.2	30.4	24.3	-	-
Sentence-Condition (Zhou et al., 2016)	72	54.6	40.4	29.8	24.5	-	95.9
LSTM (Vinyals et al., 2015)	68.4	51.2	38	28.4	23.1	50.7	84.3
LSTM-A$_1$	72.3	55.8	42	31.7	24.9	53.3	96
LSTM-A$_2$	72.8	56.4	42.7	32.2	25	53.5	97.5
LSTM-A$_3$	**73.1**	56.4	42.6	32.1	**25.2**	53.7	98.4
LSTM-A$_4$	71.1	54.5	40.9	30.6	24	52.5	90.6
LSTM-A$_5$	73	**56.5**	**42.9**	**32.5**	25.1	**53.8**	**98.6**

- Sentence-Condition (Zhou et al., 2016): Sentence-condition is proposed most recently and exploits text-conditional semantic attention to generate semantic guidance for sentence generation by conditioning image features on current text content.

- MSR Captivator (Devlin et al., 2015): MSR Captivator employs both Multimodal Recurrent Neural Network (MRNN) and Maximum Entropy Language Model (MELM) (Fang et al., 2015) for sentence generation. Deep Multimodal Similarity Model (DMSM) (Fang et al., 2015) is further exploited for sentence re-ranking.

- CaptionBot (Tran et al., 2016): CaptionBot is a publicly image captioning system[2] which is mainly built on vision models by using Deep residual networks (ResNets) (He et al., 2016) to detect visual concepts, MELM (Fang et al., 2015) language model for sentence generation and DMSM (Fang et al., 2015) for caption ranking. Entity recognition model for celebrities and landmarks is further incorporated to enrich captions and the confidence scoring model is finally utilized to select the output caption.

- LSTM-A: LSTM-A$_1$, LSTM-A$_2$, LSTM-A$_3$, LSTM-A$_4$, and LSTM-A$_5$ are five variants derived from our proposed LSTM-A framework.

4.4 PERFORMANCE COMPARISON

Performance on COCO   Table 1 shows the performances of different models on COCO image captioning dataset. Overall, the results across seven 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 can achieve 98.6%, which is to date the highest performance reported on COCO dataset 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 11.6%, 7.8%, 5.1%, and 13.9% in BLEU@4, METEOR, ROUGR-L, and CIDEr-D, 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 quickly 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 (`http://mscoco.org/dataset/#captions-leaderboard`), 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
MSM@MSRA (LSTM-A$_3$)	**73.9**	**91.9**	**57.5**	**84.2**	**43.6**	**74**	**33**	**63.2**	**25.6**	**35**	**54.2**	**70**	**98.4**	**100.3**
ATT (You et al., 2016)	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 (Vinyals et al., 2015)	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 (Devlin et al., 2015)	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

Figure 2: Attributes and sentences generation results on COCO. The attributes are predicted by MIL method in (Fang et al., 2015) and the output sentences are generated by 1) LSTM, 2) CaptionBot[2], 3) our LSTM-A$_3$, and 4) Ground Truth: randomly selected three ground truth sentences.

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, is benefited from the mechanism of first feeding 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 five evaluation metrics.

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 in our final submission and only the latest top-3 performing methods which have been officially published are included in the table. Compared to the top performing methods, our proposed LSTM-A$_3$ achieves the best performance across all the evaluation metrics on both c5 and c40 testing sets, and to-date ranks the first on the Leaderboard. In addition, when training the attribute detectors by ResNet-152, our CIDEr-D scores on c5 and c40 testing sets will be further boosted up to 104.9% and 105.3%, respectively.

Figure 3: The effect of beam size k on (a) LSTM-A$_3$ and (b) LSTM-A$_5$.

(a) k for LSTM-A$_3$

(b) k for LSTM-A$_5$
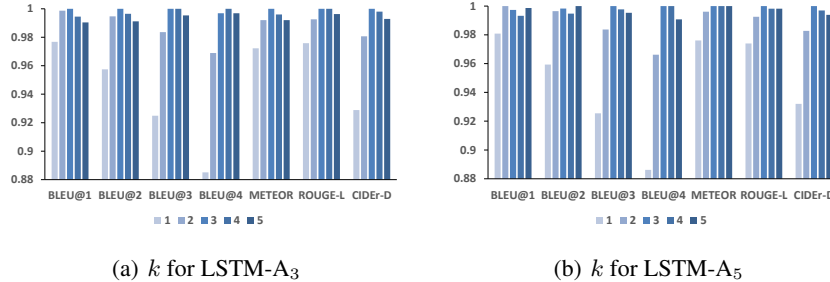
4.5 QUALITATIVE ANALYSIS

Figure 2 shows 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$_3$ can 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.

4.6 ANALYSIS OF THE BEAM SIZE k

In order to analyze the effect of the beam size k in testing stage, we illustrate the performances of our two top performing architectures LSTM-A$_3$ and LSTM-A$_5$ with the beam size in the range of $\{1, 2, 3, 4, 5\}$ in Figure 3. To make all performances fall into a comparable scale, all scores are normalized by the highest score of each evaluation metric. As shown in Figure 3, we can see that almost all performances in terms of each evaluation metric are like the "$\wedge$" shapes when beam size k varies from 1 to 5. Hence, we set the beam size k as 3 in our experiments, which can achieve the best performance with a relatively small beam size.

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 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 clearly observed when comparing to other captioning techniques and more remarkably, the performance of our LSTM-A to date ranks the first on COCO image captioning Leaderboard.

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. We will further analyze the impact of different number of attributes involved. Second, how to enlarge the word vocabulary of generated sentences with the learnt attributes is worth trying and seems very interesting.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. Language models for image captioning: The quirks and what works. In *ACL*, 2015.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *CVPR*, 2015.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *MM*, 2014.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *ICML*, 2014.

Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. on PAMI*, 2013.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*, 2004.

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. In *ECCV*, 2014.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Explain images with multimodal recurrent neural networks. In *NIPS Workshop on Deep Learning*, 2014.

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.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, 2011.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

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

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, and Chris Sienkiewicz. Rich image captioning in the wild. *arXiv preprint arXiv:1603.09016*, 2016.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. What value do explicit high level concepts have in vision to language problems? In *CVPR*, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

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

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

Luowei Zhou, Chenliang Xu, Parker Koch, and Jason J Corso. Image caption generation with text-conditional semantic attention. *arXiv preprint arXiv:1606.04621*, 2016.