Pattern Recognition Letters 000 (2017) 1–9

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Image Caption Generation with Part of Speech Guidance

Xinwei He[a], Baoguang Shi[a], Xiang Bai[a,*], Gui-Song Xia[b], Zhaoxiang Zhang[c], Weisheng Dong[d]

[a] School of Electronic Information and Communications, Huazhong University of Science and Technology (HUST), Wuhan 430074, China
[b] State Key Lab. LIESMARS, Wuhan University, Wuhan 430079, China
[c] Institute of Automation, CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, China
[d] School of Electronic Engineering, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Image caption generation
Part-of-speech tags
Long Short-Term Memory
Visual attributes

ABSTRACT

As a fundamental problem in image understanding, image caption generation has attracted much attention from both computer vision and natural language processing communities. In this paper, we focus on how to exploit the structure information of a natural sentence, which is used to describe the content of an image. We discover that the Part of Speech (PoS) tags of a sentence, are very effective cues for guiding the Long Short-Term Memory (LSTM) based word generator. More specifically, given a sentence, the PoS tag of each word is utilized to determine whether it is essential to input image representation into the word generator. Benefiting from such a strategy, our model can closely connect the visual attributes of an image to the word concepts in the natural language space. Experimental results on the most popular benchmark datasets, *e.g.*, Flickr30k and MS COCO, consistently demonstrate that our method can significantly enhance the performance of a standard image caption generation model, and achieve the conpetitive results.

1. Introduction

Automatically generating caption for images has become an active research topic recently in computer vision community, which tries to describe the content of a given image with a reasonable sentence in English or other languages. Specifically, to describe an image, a model for image caption generation often encodes the objects, semantic attributes, object relationships, and possible activities existing in an image, into a representation vector. Then, a word generator can be adopted on top of it to generate the corresponding caption word by word. Following this paradigm, a large amount of approaches on this task have been presented due to the recent advances in deep learning techniques [1,6,28,29].

Our work is motivated by one of the most popular frameworks following the mainstream encoder-decoder pipeline proposed by Vinyals et al. [31] which combines Convolutional Neural Networks (CNN) to obtain visual feature vector and LSTM networks [12] to decode the vector into natural language sentences. One major drawback of such a framework lies in the fact that the feature vector is only fed into the LSTM network at the initial time step, which often causes the "drift away" effect. The "drift away" effect is pointed out by Jia et al. [13], which means that the LSTM network may lose track of the original image content, being prone to fit the sentences in the training set. A straightforward solution might be adding the image feature vector as an extra input to all the units of LSTM blocks. However, such a strategy may easily lead to the LSTM model overfitting the image content, worsening the system performance. Instead, Jia et al. [13] add semantic feature vector extracted from the image as an extra input to each LSTM unit, in order to guide the LSTM model to overcome the problems of overfitting and "drift away".

Natural language sentence and visual representation of an image are two signals of very different modalities. When we try to connect a visual representation such as CNN feature to a natural language sentence, one critical problem is how to integrate the image representation and language knowledge closely within a language model. Different from previous works focusing on visual representation extraction, we consider from another perspective using the Part-of-Speech (PoS) tag of each word as an informative cue to enhance the learning of LSTM model, which is popular in natural language processing, but never used in image captioning. We notice that not all the basic atoms or words of a natural language sentence can be mapped to the corresponding parts of an image. For instance, a determiner (the, any, a, an, *etc.*) or preposition (behind, below, across, *etc.*) cannot directly correspond to any explicit region of an image. On the contrary, the image regions

* Corresponding author. Tel.: +8613297073017.
E-mail address: xbai@hust.edu.cn (X. Bai).

https://doi.org/10.1016/j.patrec.2017.10.018
0167-8655/

(a) caption: A cat watching a bird on a tv. PoS tags: DT NN VBG DT NN IN DT NN.

(b) A wooden table topped with green juice and with fruits and vegetables. PoS tags:DT JJ NN VBN IN JJ NN CC IN NNS CC NNS.

Fig. 1. Two exemplar images from MS COCO [4]. The sentences below the two images are their corresponding reference captions with the PoS tag for each word. The words and their corresponding PoS tags marked in red color, are closely related to the image content.

including objects, their attributes and activities that usually make sense to caption generation, often correspond to nouns, verbs, and adjectives of a natural language sentence, respectively. As a consequence, in this paper we follow the abbreviations of PoS labels in natural language. To clearly illustrate our motivation, two sentence examples as well as the PoS tag of each word are shown in Fig. 1 . When a word belongs to the set of NN, VB, JJ, we can explicitly connect it to the image content. On the other hand, we cannot easily connect the image content to the words belonging to DT, CC, IN. Motivated by this observation, we propose a novel strategy for learning the LSTM model according to the structure information of the training sentences. Specifically, the PoS tag of each word is acting the role of a switch signal, which is used to determine whether to input the visual feature to the LSTM block. This idea is reasonable, as a machine only needs to look at the image for describing the objects, attributes, and activities.

Our main contributions are two-fold. First, as mentioned above, using PoS tags to train an image caption generator enables the model to connect natural language sentences and image content more closely. Second, to overcome the overfitting issue, we use the combination of visual feature vector and the word embedding vector as the input to each LSTM unit. We simply add the image feature vector extracted by CNN to the word vector, weakening the influence of overfitting on the image content caused by LSTM. In our experiments, we demonstrate that PoS tags have a positive impact on the performance of the whole image captioning system. Under the guidance of PoS tags, we can achieve the state-of-the-art results on the benchmark datasets with our image captioning framework.

2. Related Work

An image captioning system needs to not only understand images, but also express them in grammatically-correct and human-readable sentences. Thus, image captioning is a cross-domain problem, highly relevant to computer vision and natural language processing. A comprehensive survey is provided by Bernardi et al. in [3].

Most recent approaches on image caption generation can be categorized as either *retrieval-based* or *generation-based*. Retrieval-based approaches address such a problem by retrieving similar images in the training dataset according to some similarity metric, *e.g.* the Euclidean distance between the extracted feature vectors. Finally, the captions of the candidate images are ranked and the best candidate caption is transferred to the input image. For instance, Ordonze et al. [23] create a web-scale captioned image dataset, from which a set of candidate matching images are retrieved out using their global image descriptors. Then, all images in the matching set are re-ranked. Finally, the best caption for the query image is returned. More recently, Kiros et al. [16] propose an encoder-decoder pipeline to learn multimodal joint space for images and text, and use the embedded features to retrieve the properest sentence to describe the query image.

Generation-based approaches adopt a totally different scheme: directly generating captions from images with a learned model, usually a deep learning model. For instance, Vinyals et al. [31] employ the output of the fully connected layer of CNN as visual feature of the image and directly feed it into the LSTM to generate a sentence describing the input image. On the other hand, Xu et al. [33] consider the output feature maps from a lower convolutional layer as visual features. And an attention mechanism is adopted for iteratively selecting a spaital feature vector as the input of the LSTM. Furthermore, Jia et al. [13] modify the LSTM and introduce semantic features as the guidance to the LSTM. By directly employing high-level attributes to guide the language model, You et al. [35] and Wu et al. [32] successfully advance the automatic metrics like CIDEr and BLEU in image caption. In addition, Yao et al. [34] study various schemes of fusing image representation and attributes to further boost the image captioning performance. Pan et al. [24] use the transferred semantic attributes mined from images and videos, and then leverage them by a novel transfer unit before feeding them to LSTM for video caption. Following the similar idea that the model does not need visual input at very time steps when generating the captions, Lu et al. [19] propose to use a visual sentinel in the decoder to decide when to attend to the image. More recently, Liu et al. [18] propose to use policy gradient optimization to directly optimize for evaluation metric like CIDEr [30] of the image caption model. They have improved state-of-the-art by a large margin. Other works of this kind include [5,10,11].

Generation-based approaches are capable of generating novel sentences from never-seen images, thus more flexible.

Our work belongs to the generation-based category. We enhance the popular CNN-LSTM model [31] by incorporating PoS tags into the generation process. During the training phase, we guide the LSTM with the PoS tag information of caption words. This has the effect of making the LSTM aware of the grammatical category of the current word to a degree. And during the testing phase, we sample the PoS tag of the current word from the lookup table which stores the grammatical category of each word. Since our model neither relies on the retrieved semantics like [13] nor uses the complex and expensive attention mechanism like [33], it can be easily trained and preserve considerable accuracy meanwhile.

3. Proposed Model

Our model is based on the prevalent *encoder-decoder* framework [6], which is flexible and effective. Following the state-of-the-art methods on image captioning [13,14,20,33], we employ a convolutional neural network (CNN) as the encoder, and a recurrent neural network (RNN) as the decoder. In this section, we first briefly introduce the whole structure of the proposed model in Section 3.1. Then, in Section 3.2, we introduce image caption generation with PoS tags in detail.

3.1. Generating captions from images

The whole framework of our model is illustrated in Fig. 2. The first part, *i.e.*, the encoder, is a convolutional neural network. Given an input image, denoted by I, the encoder extracts a high-level feature representation. Following previous works [13,14,33], we adopt the VGG16 [27] architecture as the encoder.

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

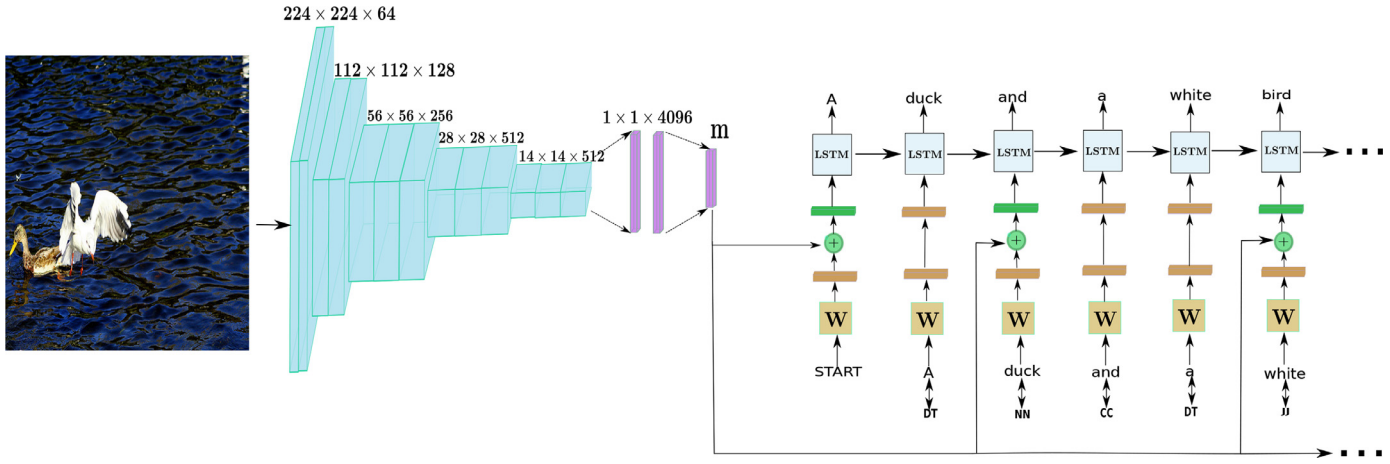


Fig. 2. Illustration of the proposed model.Our model consists of a VGG16 CNN as the encoder, and an LSTM network as the decoder. Given an input image *I*, the CNN extracts a 4096-dimensional feature vector as the image representation. The following LSTM (unrolled to multiple steps in this figure) takes the image representation and a START token, and recurrently generates the next word in the output sentence. At each step, the decoder use the switch to decide whether to feed the image feature into the LSTM, conditioned on the PoS tag of the last output word.

The original architecture is designed for image classification tasks. We remove the last fully connected layer and the softmax layer, so that the resulting CNN outputs a 4096-dimensional vector which describes the global content of an input image. Also, following the standard practice, we transfer the weights of the VGG16 model pretrained on ImageNet [26] into our model.

The decoder is a word generator. In image caption, given the image representation produced from the encoder, the decoder generates words of the output sentence one by one, in a recurrent process. Mathematically, a probability distribution $P(S|I)$ is defined over $S = <w_1, w_2, ..., w_{|S|}>$ conditioned on image I. The model is trained to maximize this posterior on a training dataset. The distribution is further factorized into:

$$P(w_1, w_2, ..., w_{|S|}|I) = \prod_{t=1}^{|S|} P(w_t|I, w_{1:t-1}), \tag{1}$$

where we assume that the generation of each individual word depends on the image I and previously generated words $w_{1:t-1}$.

The recursiveness of Eq. 1 allows us to model the distribution via a recurrent neural network (RNN). To model Eq. 1, each factor corresponds to an unrolled step of RNN inference. In this paper, we employ LSTM [12] as the RNN unit. As illustrated in Fig. 3, an LSTM unit has a memory cell state and three gates. The three gates are called the input, output, and forget gates respectively.

At a given time-step t, the LSTM receives the hidden state $\mathbf{h}_{t-1}$ and the cell state $\mathbf{c}_{t-1}$ from the last step, as well as the input $\mathbf{u}_t$ at this step. Denoting the feed-forward function as $f_{\text{LSTM}}(\cdot)$, the LSTM updates its state by:

$$\mathbf{h}_t, \mathbf{c}_t = f_{\text{LSTM}}(\mathbf{u}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}). \tag{2}$$

Internally, the LSTM undergoes a series of computations on its gates and memory cell:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{u}_t + \mathbf{R}_i\mathbf{h}_{t-1} + \mathbf{b}_i); \tag{3}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{u}_t + \mathbf{R}_f\mathbf{h}_{t-1} + \mathbf{b}_f); \tag{4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{u}_t + \mathbf{R}_o\mathbf{h}_{t-1} + \mathbf{b}_o); \tag{5}$$

$$\mathbf{z}_t = \tanh(\mathbf{W}_z\mathbf{u}_t + \mathbf{R}_z\mathbf{h}_{t-1} + \mathbf{b}_z); \tag{6}$$

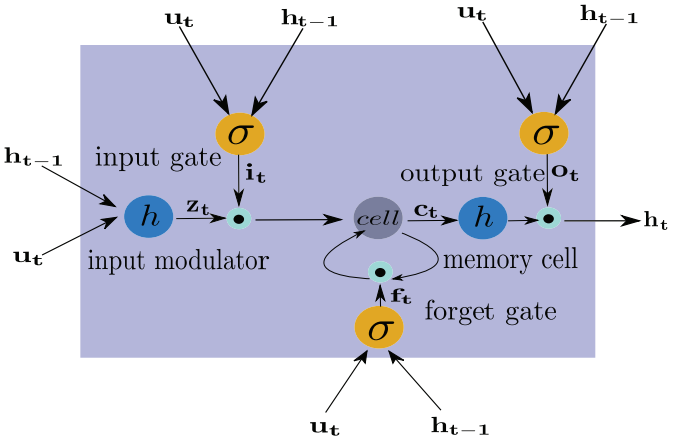


Fig. 3. The inner structure of an LSTM cell. The LSTM cell takes $\mathbf{u}_t$ and $\mathbf{h}_{t-1}$, based on which it calculates the values for three multiplicative gates, namely the input, the forget and output gates. The gates control the information that flows into and out of the cell, making LSTM capable of capturing long-term dependencies in a sequence.

$$\mathbf{c}_t = \mathbf{i}_t \odot \mathbf{z}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1}; \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{8}$$

where $\mathbf{W}_\star$, $\mathbf{R}_\star$, $\mathbf{b}_\star$ are the learned input weight matrices, recurrent weight matrices and bias vectors respectively. σ represents the sigmoid function taking the form $\sigma(x) = 1/(1+\exp(-x))$ which squashes inputs to the range of (0, 1). tanh represents the hyperbolic tangent function which squashes inputs to the range of $(-1, 1)$. These functions are applied to input vectors element-wisely. $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ represent the input gate, the forget gate, and the output gate respectively. They are computed by adding the linear projections of $\mathbf{u}_t$ and $\mathbf{h}_{t-1}$, followed by a sigmoid function. The gates respectively modulate the input transformation $\mathbf{z}_t$, the previous cell value $\mathbf{c}_{t-1}$, and the output by element-wise multiplications, denoted by $\odot$. In our experiments, the input of LSTM $\mathbf{u}_t$ at time step t can be either a word embedding or the summation of a word embedding and image features, conditioned on a switch signal, as shown in Eq. 13.

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

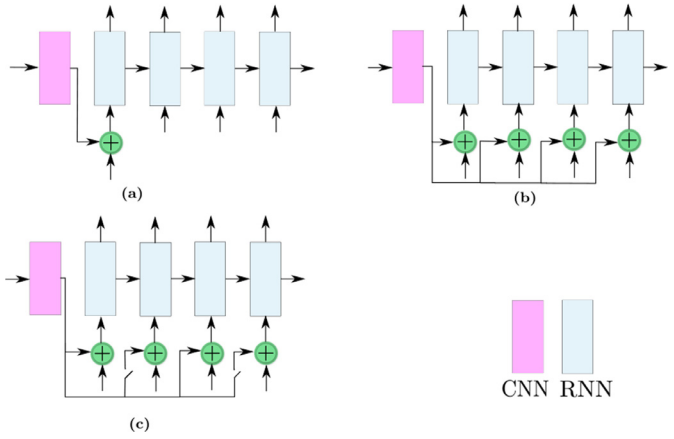


Fig. 4. Illustration of three different ways of exposing an image to a generator: (a) input the image features only at the first step; (b) input the image features at every step; (c) we propose to input the image features at some particular steps, conditioned on the PoS tag of the last word.

Following the LSTM updates, the word at time step t, denoted by w_t, is sampled from a predicted probability distribution over a dictionary:

$$w_t \sim \text{softmax}(\mathbf{D}\mathbf{h}_t). \tag{9}$$

where $\mathbf{D}$ is the learned decoding weight matrix.

The input $\mathbf{u}_t$ to the LSTM depends on both the image features and the previous word w_{t-1}. The detailed formulation will be given later in Section 3.2. Overall, the generation process can be summarized as:

$$w_t \sim P(w_t|w_1, \ldots, w_{t-1}, I; \boldsymbol{\theta}), \tag{10}$$

where $\boldsymbol{\theta}$ denotes all model parameters to be learned. As indicated by Eq. 10, each generated word depends on both the input image I and all previously generated words. The former dependency is necessary for generating description related to the image content. The latter one models the dependencies between consecutive words in a sentence, thus guaranteeing the grammatical correctness of the output sentence.

3.2. PoS guided generation

To generate meaningful descriptions for the images, a word generator must refer to the image content during the generation process. A common approach, as illustrated in Fig. 4a, is to let the generator look at the input image once at the beginning, and store the image content by projecting the image features into its memory cells. In later steps, the image features are no longer fed into the generator. Recurrent neural networks such as LSTM networks are capable of holding long-term memory, therefore even after many generation steps, the input image may still have an effect on the generated words.

Despite the ability of holding long-term memories, such word generators are still prone to the "drift away" problem, as pointed out by Jia et al. [13]. Such a generator easily losses track of the original image content, subsequently generating words only based on the word dependencies learned from training captions. Consequently, the generated captions tend to be identical or similar to training captions, limiting the generalization ability.

We argue that, in order to address the "drift away" effect, it is important to apply image features to the word generator during the generation process. To realize this, a straightforward solution is to feed image features into the word generator at every generation step, as illustrated in Fig. 4b. At step t, the input to the LSTM consists of two parts. The first one is the embedding vector of the previous word w_{t-1} (w_0 is the START token), we use the embedding function $e(\cdot)$ proposed in [21,22], which simply maps a one-hot representation $x_{t-1} \in \mathbb{R}^{|V|\times 1}$ of the word w_{t-1} to a corresponding row vector of the embedding matrix $E \in \mathbb{R}^{|V|\times d}$, where d represents the word embedding size, $|V|$ is the vocabulary size. This can be formulated as follows,

$$w_{t-1} \mapsto x_{t-1}, e(x_{t-1}) = E^T x_{t-1} \tag{11}$$

$e(x_{t-1})$ represents the word vector of w_{t-1}; $\mathbf{m}$ is the projection of the CNN features. The two vectors, $e(w_{t-1})$ and $\mathbf{m}$, are added up element-wisely before being fed into the LSTM. To formulate mathematically,

$$\mathbf{u}_t = e(x_{t-1}) + \mathbf{m}, \tag{12}$$

where $\mathbf{u}_t$ is the LSTM input vector used in Eq. 2. The resulting $\mathbf{u}_t$ incorporates information of both image and word, thus is richer and more meaningful.

Compared with the popular approach shown in Fig. 4a, the second approach, shown in Fig. 4b, allows a generator to have an explicit reference to image content during every generation step. However, the reference to image content may not be useful at all the steps. In fact, some words of a sentence may be weakly related to the image to be described. For example, in the sentence *"a man is standing on a beach with a kite"*, words including *"man", "standing", "beach", "kite"* can not be inferred without referring to the input image. Whereas, other words such as *"is", "with"* support the grammatical correctness of the sentence, but they are weakly related to the image content.

Motivated by this, we propose another generation strategy. We argue that a word generator needs the reference to image content only at *some* particular steps. Considering this, we add a special *switch* into our generator, as illustrated in Fig. 4c. When the switch is turned on, image features are fed into the sum module, whose another branch is the word embedding $e(x_{t-1})$. The generator thus receives the fused feature of both image and word embedding. Otherwise, if the switch is turned off, the generator only receives the word embedding.

We find the necessity of feeding image feature (or equivalently the on-off status of the switch) at the current step according to the grammatical attribute of the previous word. Therefore, we make the switch controlled by the grammatical category of w_{t-1}. Specifically, we use the PoS tags generated by a syntactic parser, Senna [7,9]. Senna classifies each word into one of its predefined *tags*, including NN, VB, JJ, *etc.*, called PoS tags. From the full set of PoS tags we select a subset (*tagset*), denoted by Ω. If the PoS tag of w_{t-1} is in the selected tagset, the switch is turned on, otherwise off.

Putting them together, we formulate the input to the LSTM, i.e. $\mathbf{u}_t$, as:

$$\mathbf{u}_t = e(x_{t-1}) + \mathbf{m} \odot \mathbb{1}(\text{PoS}(w_{t-1}) \in \Omega \cup \{\text{START}\}), \tag{13}$$

where $\text{PoS}(\cdot)$ maps a word to its PoS tag. $\mathbb{1}(\cdot)$ is an indicator function. If the PoS tag is not in $\Omega \cup \{\text{START}\}$, the function outputs a zero vector whose size is the same as $\mathbf{m}$, thus excluding image features from $\mathbf{u}_t$. Notice that we include the special token START in the set, so that at the first step the LSTM always receives image feature.

3.3. Training and Inference

The training process aims to maximize the log-likelihood of the caption sentences conditioned on the input images. According to Eq. 1, the loss function is formulated as: *i.e.*

$$\mathcal{L} = \sum_{I,S\in\mathcal{X}} \log P(S|I; \boldsymbol{\theta}) \tag{14}$$

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

$$= \sum_{I,S\in\mathcal{X}} \sum_{t=1}^{|S|} \log P(w_t|w_{1:t-1}, I; \boldsymbol{\theta}), \quad (15)$$

where $S = \{w_1, w_2, \ldots, w_{|S|}\}$ is a caption sentence in the training dataset $\mathcal{X}$; $P(w_t|w_{1:t-1}, I; \boldsymbol{\theta})$ is the conditional probability of the current word w_t, which depends on the input image I, the model parameters $\boldsymbol{\theta}$, and the previous generated words context $w_{1:t-1}$.

Different from the training procedure, we do not have access to the whole caption during the testing procedure. Therefore, an iterative strategy is adopted. At each step, the word generator samples a word from the predicted distribution, and feeds the word in the next time step. Once the special token EOS is sampled, the iterative process terminates.

A straightforward word sampling strategy is to pick out the word with the highest conditional probability at every step. However, a beam search based strategy gives better performance. Instead of picking out only one word, we store the top-k words at every step. k is called the beam width. In the end, we trace back from the last word to the first one, selecting the best path of word sequence. The beam search based strategy boosts the performance by 1% to 2% in terms of BLEU score.

Also, during test, the precise PoS tag of each word is unavailable, because they could only be inferred when the sentence is complete. However, we find that most words have only one corresponding PoS tag. Therefore, we construct a dictionary that maps each word to its PoS tag (or tags). If any tag of a word is in the pre-selected tagset Ω, the switch is turned on.

4. Experiments

In this section, we will first introduce the experimental settings in three aspects: the datasets we used, the evaluation metrics, and the implementation details. Then, we evaluate the performance of our model on the standard benchmark datasets, and compare the proposed method with recent state-of-the-art methods. Finally, we give a discussion on the choice of the PoS tagset Ω.

4.1. Datasets

To verify the effectiveness of our model, we conduct experiments on two popular and challenging datasets: Flickr30K [36] and MS COCO [4].

Flickr30K consists of 31,783 images, each annotated with 5 reference captions by crowd-sourcing. Hence, it contains 158,915 captions in total. This dataset is collected from Flickr[1]. Most images of this dataset are about human activities in daily life. Following previous works such as [14,33], we randomly pick 1,000 images for validation, 1,000 for testing, and the rest for training.

MS COCO is a comparatively large-scale dataset released by Chen et al. in [4]. This dataset contains 164,062 images with captions. Unlike Flickr30K, MS COCO gives an official training set split which contains 82,783 images and 413,915 captions, and a validation set which contains 40,504 images and 202,520 captions. Each training/validation image has 5 captions. MS COCO also has a test set whose ground truth captions are not publicly available. Following previous works including [13,14,33], we train our model on the whole training set, and evaluate its performance on a validation subset which contains 5,000 images.

4.2. Evaluation Metrics

In our experiments, we adopt three kinds of metrics including BLEU [25], METEOR [2], and CIDEr [30], in order to provide a comprehensive evaluation of our model. In general, the higher scores a model achieves under these metrics, its generated captions are more similar to the human-annotated reference captions.

BLEU [25] stands for Bilingual Evaluation Understudy. It is a metric that has been extensively used for evaluating machine translation algorithms. BLEU is designed to evaluate the quality of corpuses, but it performs poorly for evaluating sentences.

METEOR [2] is the abbreviation of Metric for Evaluation of Translation with Explicit ORdering. It is another metric for evaluating the alignment between words in the candidate and reference sentences.

CIDEr [30] stands for Consensus-based Image Description Evaluation. This metric is specifically designed for evaluating image captioning algorithms.

4.3. Implementation details

The CNN in our model follows the original VGG16 [27] architecture, which is pretrained on ImageNet [26]. Besides, we connect a linear layer that projects the vector into one with its size equal to the word embedding vector so that they can be summed element-wise directly. The new linear projection layer is randomly initialized.

For the word generator, we adopt a single-layer LSTM. On the two datasets, we use different numbers of LSTM hidden units. The LSTM is set with 256 and 512 hidden units for the experiments on Flickr30K and MS COCO, respectively. The weights of the LSTM are randomly initialized from a Gaussian distribution which has zero mean and 0.08 deviation.

Our model is trained with Adam [15] algorithm, which works best among the popular optimization techniques in our experiments. For stability, we set different learning rates for the CNN and the LSTM. Specifically, we use a learning rate of 10^{-4} for the LSTM network. The learning rate of the CNN is set to zero for the first 5,000 training iterations, then set to 10^{-5}. The mini-batch size is set to 16 for all the experiments. During the testing procedure, the beam width is set to 2, 8 for MS COCO and Flickr30K, respectively.

We use the latest version of Senna[2] for generating PoS tags. Senna generates nearly 40 different types of PoS tags. The full tagset is denoted by $\mathcal{S} = \{\text{NN, DT, IN, JJ, ...}\}$. The tagset Ω in Eq. 13 is a subset of $\mathcal{S}$. In our experiments, we choose $\Omega = \mathcal{S} - (\{\text{DT, CC, TO}\} \cup \Upsilon)$, where Y contains PoS tags with the lowest frequencies on MS COCO. The choice of PoS tagset will be discussed in Section 4.5.

The proposed model is implemented under the GPU-accelerated framework Torch7 [8]. We run the proposed algorithm on a workstation with two 2.40 GHz 6-core Intel CPUs, one GeForce GTX 780 graphics cards, and 64GB physical memory. The training process takes about 2 days on Flickr30K and 3.5 days on MS COCO respectively. Testing an image takes less than 0.1 second.

4.4. Benchmark results

We compare our model with several state-of-the-art models for image caption, such as Google NIC [31], Guiding LSTM [13], *etc.* on Flickr30K and MS COCO datasets. The results of our model as well as other methods on Flickr30K and MS COCO datasets are shown in Table 1. The performance of our model is very competitive among the state-of-the-art methods on Flickr30K. In particular, our model achieves state-of-the-art results on both BLEU-3 and BLEU-4, outperforming the recent Guiding LSTM model [13]. Our model increases the BLEU-3 and BLEU-4 by nearly 3 points compared with the strong baseline Google NIC [31].

[1] https://www.flickr.com/.

[2] Downloaded and compiled from https://github.com/torch/senna.

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

Table 1
The performance comparisons between our method and recent state-of-the-art methods on Flickr30k and MS COCO, "†" means different data splits[3], "Δ" indicates using model ensembles technique, "-" indicates unknown metrics.

Flickr30k				MS COCO						
Methods	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	METEOR	CIDEr
nearest neighbour(base line)	-	-	-	-	0.480	0.281	0.166	0.100	0.157	0.383
Google NIC† Δ [31]	0.663	0.423	0.277	0.183	0.666	0.461	0.329	0.246	-	-
Karpathy&Fei-Fei [14]	0.573	0.369	0.240	0.154	0.625	0.450	0.321	0.230	0.195	0.660
LRCN † [10]	0.587	0.391	0.251	0.165	0.669	0.489	0.349	0.249	-	-
Phased-based IC [17]	0.59	0.35	0.20	0.12	0.70	0.46	0.30	0.20	-	-
m-RNN† [20]	0.60	0.41	0.28	0.19	0.67	0.49	0.35	0.25	-	-
Soft-Attention [33]	0.667	0.434	0.288	0.191	0.707	0.492	0.344	0.243	**0.239**	-
Hard-Attention [33]	**0.669**	0.439	0.296	0.199	**0.718**	0.504	0.357	0.250	0.230	-
Guiding LSTM [13]	0.646	**0.446**	0.305	0.206	0.67	0.491	0.358	0.264	0.227	0.813
Our proposed model	0.638	**0.446**	**0.307**	**0.211**	0.711	**0.535**	**0.388**	**0.279**	**0.239**	**0.882**
State-of-the-art results using extra attributes information										
Att-CNN+LSTM [32]	0.73	0.55	0.40	0.28	0.74	0.56	0.42	0.31	0.26	0.94
ATT-FCN [35]	0.647	0.460	0.324	0.230	0.709	0.537	0.402	0.304	0.243	-
LSTM-A5 [34]	-	-	-	-	0.73	0.565	0.429	0.325	0.251	0.986

[3] On MS COCO, NIC reserves 4K images for testing, m-RNN samples 4K images for validation and 1 K images for testing, LRCN isolates 5K images from validation set for testing.

Table 2
The performance comparisons between our model and some state-of-the-art models on the MSCOCO evaluation server.

Methods	B-1		B-2		B-3		B-4		METEOR		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Attention [33]	0.705	0.881	0.528	0.779	0.383	0.658	0.277	0.537	0.241	0.322	0.865	0.893
m-RNN [20]	0.716	0.890	0.545	0.798	0.404	0.687	0.299	0.575	0.242	0.325	0.917	0.935
Att-CNN+LSTM [32]	0.725	0.892	0.556	0.803	0.414	0.694	0.306	0.582	0.246	0.329	0.911	0.924
Our Proposed Method	**0.728**	**0.904**	**0.560**	**0.819**	**0.420**	**0.715**	**0.313**	**0.603**	**0.249**	**0.336**	**0.942**	**0.951**

In terms of BLEU-3, 4, our model performance is better than Guiding LSTM model [13]. Though the results on BLEU-1 is poorer than Google NIC [31], they are still comparable.

With more training data, our model achieves even better results. On MS COCO dataset, our model outperforms the other state-of-the-art methods by nearly 3 points on average in terms of BLEU-1, 2, 3, 4 scores. In terms of METEOR, the performance of our model is on par with the soft-attention model proposed in [33], which adopts a more sophisticated attention-based network architecture. Comparing to Google NIC [31], our model performs much better in terms of BLEU-1 4 metrics, for example, nearly 3 points improvement on BLEU-4 metric. Moreover, compared with the recent Guiding LSTM model [13], our model also shows a significantly better performance. Overall, our model consistently achieves state-of-the-art performance on most of the metrics, further demonstrating its effectiveness and generalization capability.

In addition, we also do an online test on the MS COCO server. And the results are shown in Table 2. As can be see from the table, our model has achieved better results than some of the state-of-the-art methods on MS COCO official test set. Comparing to **Att-CNN+LSTM** [32], our models increase CIDER-D metric by almost 3 points.

In Fig. 5, we show some example captions generated by our model on MS COCO dataset. Generally, our model is able to capture the essential meanings in images, and generate human-readable captions. For example, the generated caption of Fig. 5l shows that the model captures the main objects in the image, *i.e.*, zebra, trees, and field. Moreover, it understands some actions, such as *"saying"* and *"standing"*, which belong to high-level concepts.

Our model fails under some cases, particularly when backgrounds are complex. Though our model mostly understands the major content of an image, it sometimes falsely counts the number of objects. In Fig. 5h, the generated caption tells the wrong number of zebras. And as shown in Fig. 5i, Fig. 5j and Fig. 5l, when the color of the object is similar to the background, the proposed model may make mistakes on the object's activity and the object category. And also in Fig. 5k, even though our model produces relevant caption, Some mistakes on gender and age have been made. We assume that the global visual feature extracted from the image alone is not enough to guide the language model when the background is complex. More fine-grained visual features need to be used to avoid such mistakes.

4.5. Discussion

A key contribution of this paper is to adopt PoS tags to guide the caption generation process. Therefore, the choice of the PoS tagset Ω is important. Notice that the PoS tag of the current word controls whether to feed image features when predicting the *next* word. Unfortunately, we are unable to know the optimal choice of Ω in prior. Therefore, we first use Flickr30K for our ablation studies on the performance impact of each PoS category. Then guided by the ablation analysis results, we manage to find the best Ω by testing several different combinations empirically.

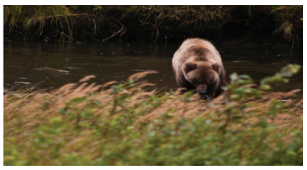
Table 3 summarizes the ablation results on the contributions of every single PoS tag and several different combinations of PoS tags to our model. Senna tool's output space contains nearly 40 distinct tags. However, we observe that most tags rarely occur in the caption corpus. So to simplify the experiment, we choose the top 6 most frequent tags to do our ablation experiments, which will account for nearly 90 percent of the caption corpus. In addition, we also perform some preprocessing operations like union on the PoS tags. For example, we suppose that different variations of verb such as VBZ, VBN, *etc.* to be the same grammatical category VB, and similarly, for NN tag may represents plural noun(NNS) or NN. In Table 3, we investigate the contributions of different PoS tags to the performance of our model. The baseline model is the one whose Ω is an empty set $\varnothing$. According to Eq. 13, this is equivalent to feeding image features into the LSTM only at the beginning. As

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

(a) A train is traveling down the tracks near a train.
A cargo train that is traveling down the tracks

(b) A man is holding a tennis racket on a tennis court.
Man holding tennis racket and ball in air slightly above his hand.

(c) A man is standing on a beach with a kite.
A man flying a kite at the beach while a child and another adult watch.



(d) A bear is standing in the water near a river.
A brown bear standing in the water and looking at something.

(e) A man riding a skateboard on a ramp.
A person riding a skate board on a rail.

(f) A stop sign on a street corner with a stop sign.
A stop sign has several other signs on it.

(g) A group of giraffes are standing in the grass.
Two giraffes are standing in a brown field.

(h) Two zebras standing in a field with trees.
Three zebras standing next to each other in a field.

(i) A cat laying on a bed with a blanket.
Two small whippet dogs sleeping on a messy bed.

(j) A large clock is sitting on a shelf.
A fancy bed room with a canopy and a chandelier

(k) two women sitting on a bench next to a bench.
A boy and a girl sitting on a sidewalk eating.

(l) A man is holding a baby elephant in a zoo.
A little boy sitting next to an elephant with a long trunk.

Fig. 5. Sample captions on the MS COCO dataset. Sentences in black color are captions generated by the proposed model. Sentences in the red color are reference captions annotated by human. The last row are failure cases.

Table 3
Results on Flickr30K with different choices of Ω.

Choice of Ω	B-1	B-2	B-3	B-4
$\varnothing$	0.578	0.379	0.248	0.165
{DT}	0.592	0.394	0.262	0.176
{IN}	0.598	0.404	0.265	0.174
{CC}	0.616	0.423	0.285	0.192
{JJ}	0.624	0.428	0.289	0.196
{NN}	0.631	0.436	0.297	0.201
{VB}	0.624	0.431	0.295	0.201
{DT, IN, CC}	0.624	0.431	0.295	0.201
$\mathcal{S}$	0.631	0.439	0.3	0.204
$\mathcal{S} - \{DT, CC\}$	0.633	0.443	0.306	0.210
$\mathcal{S} - (\{DT, IN, CC\} \cup \Upsilon)$	**0.638**	**0.446**	**0.307**	**0.211**
several additional baselines				
dropout images (P=0.5)	0.612	0.415	0.279	0.189
dropout images (P=P(DT))	0.625	0.432	0.294	0.197
word to word_PoS	0.63	0.437	0.3	0.205
bigram of DT	0.614	0.419	0.28	0.187

we can see from Table 3, 'NN', which stands for noun, is the best single candidate among all the PoS tags to guide the image caption model in terms of BLEU scores. In contrast, the worst is the PoS tag 'DT', which represents determiner. The performance gap is nearly 4 points in BLEU-1 $\sim$ 3 and 2 points in BLEU-4. Further, from the results, the candidates for guiding the LSTM can be ranked according to their contributions: NN > VB > JJ > CC > IN > DT. This verifies our hypothesis that choosing the tag which may be more related to the image content can result in a better model. Note that even when the PoS tag is DT, it can still improve the model performance by a margin over the baseline model. We assume that the image content information will be enhanced when the current word's grammatical category is any PoS tag. However, the problem of overfitting will arise if we ignore the PoS tags, which means that we choose to input the image feature at every time step, thus it will hurt the performance as implied by Table 3. Compared with different combinations of PoS tags, it is obvious to see that the baseline model achieves the worst performance. On average, the performance is 3% to 5% lower than the other variants in terms of BLEU score. This is very likely to be caused by the "drift away" problem, since image features are fed into the decoder only once.

Next, we test the performance of $\Omega = \mathcal{S}$, *i.e.*, feeding image features at every generation step. This scheme works much better and gives comparable results to the best results we have achieved. However, this scheme is still not optimal. We further remove some tags from Ω, namely DT, CC and TO. these tags correspond to determiner, coordinating conjunction, and "to" respectively. Since words of these tags lack for information describing the correlation between the caption sentence and image content, there is no need to reserve them in the PoS tagset to guide the generation model. As shown in Table 3, removing these tags further improves results on all metrics.

Furthermore, we investigate four other additional baseline experiments on Flickr30K and the results are shown in the bottom rows of Table 3. The first two experiments randomly dropout image features with dropout rate of 0.5 and the distributional ratio of DT (which is 0.186) respectively. As we can see that using dropout

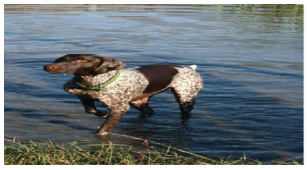
Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

(a) Ω_0 : A dog runs through the grass.
Ω_1: A dog runs through the woods.
Ω_2: A dog is walking through a grassy field.
Ω_3: A person is walking through a field of grass.
Ground truth: A tan dog walks along a grassy path with his long pink tongue hanging out.

(b) Ω_0: A man in a blue shirt is playing a game..
Ω_1: A young boy in a white shirt is playing a game.
Ω_2: Two girls are playing soccer.
Ω_3: Two young girls are playing in the air.
Ground truth: Two girls play on a fenced in field.

(c) Ω_0: A man in a blue shirt is walking down the street.
Ω_1: Two people are walking down the street.
Ω_2: Two women are standing on a sidewalk.
Ω_3: A woman in a white shirt and jeans is walking down the street.
Ground truth: People are walking in the park.

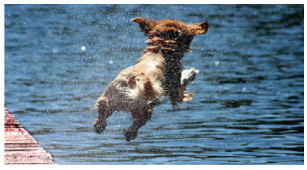
(d) Ω_0: A group of people are walking down the street.
Ω_1: A group of people are standing on a sidewalk.
Ω_2: A group of people are standing in front of a building.
Ω_3: A group of people are standing in front of a building.
Ground truth: A group of young adults spray paint large art boards..



(e) Ω_0: A black dog is running through the water.
Ω_1: A dog is running through the water.
Ω_2: A black dog is running through the water.
Ω_3: A black dog is standing in the water.
Ground truth: Black and white dog wades in water.

(f) Ω_0: A man in a blue shirt is sitting on a bench.
Ω_1: A little girl in a white shirt is standing in a market.
Ω_2: A woman is standing in front of a fruit stand.
Ω_3: A group of people are sitting in front of a fruit.
Ground truth: Woman stands at a table full of vegetable produce.

(g) Ω_0: A man in a blue shirt is sitting in the water.
Ω_1: A woman in a red shirt is holding a drink.
Ω_2: A woman in a blue shirt is standing in front of a large body of water.
Ω_3: A woman in a pink shirt is standing on a bench.
Ground truth: A woman sitting on a lawn chair.



(h) Ω_0: A black dog is running through the water.
Ω_1: A dog runs through the water.
Ω_2: A black dog is jumping into the water.
Ω_3: A black dog is running through the water.
Ground truth: A dog jumps into the water.

Fig. 6. Sample captions on the Flickr30K dataset using different selection of PoS tag sets. Sentences in the black color are captions generated by the proposed models. Sentences in the red color are reference captions annotated by human. Each PoS tagset is represented by Ω with a subscript index as following $\Omega_0 = \varnothing$, $\Omega_1 = \{\text{DT, IN, CC}\}$, $\Omega_2 = \mathcal{S} - (\{\text{DT, IN, CC, TO}\} \cup \Upsilon)$, , $\Omega_3 = \mathcal{S}$. Here we demonstrate captioning results samples of the models using 4 combinations of PoS tags. The model trained using Ω_2 generates slightly better captions than the others on Flickr30K datasets.

rate of value 0.5 performs worse than using the dropout rate of P(DT), and both have lower BLEU scores than the experiment when $\Omega = \mathcal{S}$. For the third experiment, we first preprocess each caption word with its PoS tag as its suffix and then use the transformed caption to train our model. e.g., 'a dog in a field' becomes 'a_DT dog_NN in_IN a_DT field_DT.'. For this experiment, we get a new vocabulary of size 12667 in comparison with original vocabulary size of 8612. We notice that this result is very close to the experiment when $\Omega = \mathcal{S}$. This means that just extending each word with its PoS tag in the caption does not benefit the final performance. For the fourth baseline experiment, we first do the bigram analysis of the caption data, and then feed in the image features with probability P(NN | DT) after the model sees a DT word. Based on our experiment, 71.3 percent of words after DT belong to NN, 26.5 percent are JJ and only 2 percent of words belong to other PoS tags. This baseline gives comparable results as the one which randomly dropout images with probability being 0.5. We assume that the strategy of randomly dropout images may weaken the association between the visual features and the Noun (mostly corresponding to salient objects in an image) in natural language. The best choice of Ω we found is shown in the sixth line from the bottom row of Table 3. Besides DT, CC, TO, and IN, we remove a subset Y from $\mathcal{S}$. The subset contains 20 PoS tags that have the lowest frequencies on Flickr30K. In practice, this set gives the best performance.

To demonstrate the captioning capacity of resulting models which are trained using different combinations of PoS tags as guidance, we provide some generated caption samples on Flickr30K in Fig. 6. From these examples, it is easy to see that using tag set $\Omega_2 = \mathcal{S} - (\{\text{DT, IN, CC, TO}\} \cup \Upsilon)$ as guidance usually gives us better captioning results than other models including our baseline model where the empty tag set Ω_0 is employed. It's hard for the baseline model to discriminate genders and activities, as shown in Fig. 6g and Fig. 6f *etc*. Through these comparisons, we can conclude that by feeding image features at some particular steps, these errors can be avoided to some extent.

5. Conclusion

In this paper, we propose a novel method for image caption generation that utilizes the PoS tags to guide the training and testing process. Moreover, we propose to fuse the extracted image features with the related word embeddings to avoid the overfitting to image content. The competitive results on the benchmark datasets demonstrate that Part-of-Speech guidance is effective in learning an Long Short-Term Model for such a fundamental task. In our future work, we will try our PoS guidance scheme on more sophisticated attention-based models like [33], and study more effective methods for fusing image content and word embedding.

Acknowledgments

This research was supported in part by National Natural Science Foundation of China (NSFC) (No. 61573160).

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018

References

[1] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate (2014), arXiv preprint arXiv:1409.0473.
[2] S. Banerjee, A. Lavie, in: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, 2005.
[3] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, B. Plank, Automatic description generation from images: A survey of models, datasets, and evaluation measures (2016), arXiv preprint arXiv:1601.03896.
[4] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft coco captions: Data collection and evaluation server (2015), arXiv preprint arXiv:1504.00325.
[5] X. Chen, C. Lawrence Zitnick, Mind's eye: A recurrent visual representation for image caption generation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2422–2431.
[6] K. Cho, B. Van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734.
[7] R. Collobert, Deep learning for efficient discriminative parsing, in: International Conference on Artificial Intelligence and Statistics, in: EPFL-CONF-192374, 2011.
[8] R. Collobert, K. Kavukcuoglu, C. Farabet, Torch7: A matlab-like environment for machine learning, in: BigLearn, NIPS Workshop, in: EPFL-CONF-192376, 2011.
[9] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, The Journal of Machine Learning Research 12 (2011) 2493–2537.
[10] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2625–2634.
[11] H. Fang, S. Gupta, F. Iandola, R.K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J.C. Platt, et al., From captions to visual concepts and back, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1473–1482.
[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
[13] X. Jia, E. Gavves, B. Fernando, T. Tuytelaars, Guiding the long-short term memory model for image caption generation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2407–2415.
[14] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.
[15] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014).
[16] R. Kiros, R. Salakhutdinov, R.S. Zemel, Unifying visual-semantic embeddings with multimodal neural language models (2014), arXiv preprint arXiv:1411.2539.
[17] R. Lebret, P.O. Pinheiro, R. Collobert, Phrase-based image captioning (2015), arXiv preprint arXiv:1502.03671.
[18] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, K. Murphy, Optimization of image description metrics using policy gradient methods (2016), arXiv preprint arXiv:1612.00370.
[19] J. Lu, C. Xiong, D. Parikh, R. Socher, Knowing when to look: Adaptive attention via a visual sentinel for image captioning (2016), arXiv preprint arXiv:1612.01887.
[20] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, Deep captioning with multimodal recurrent neural networks (m-rnn) (2014), arXiv preprint arXiv:1412.6632.
[21] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, S. Khudanpur, Recurrent neural network based language model., in: INTERSPEECH, 2, 2010, p. 3.
[22] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
[23] V. Ordonez, G. Kulkarni, T.L. Berg, Im2text: Describing images using 1 million captioned photographs, in: Advances in Neural Information Processing Systems, 2011, pp. 1143–1151.
[24] Y. Pan, T. Yao, H. Li, T. Mei, Video captioning with transferred semantic attributes (2016), arXiv preprint arXiv:1611.07675.
[25] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Meeting on Association for Computational Linguistics, 2002, pp. 311–318.
[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252, doi:10.1007/s11263-015-0816-y.
[27] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2014), arXiv preprint arXiv:1409.1556.
[28] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.
[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[30] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: Consensus-based image description evaluation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4566–4575.
[31] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156–3164.
[32] Q. Wu, C. Shen, L. Liu, A. Dick, A. van den Hengel, What value do explicit high level concepts have in vision to language problems, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016.
[33] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention (2015), arXiv preprint arXiv:1502.03044.
[34] T. Yao, Y. Pan, Y. Li, Z. Qiu, T. Mei, Boosting image captioning with attributes (2016), arXiv preprint arXiv:1611.01646.
[35] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016.
[36] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions, Transactions of the Association for Computational Linguistics 2 (2014) 67–78.

Please cite this article as: X. He et al., Image Caption Generation with Part of Speech Guidance, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.018