Reference Based LSTM for Image Captioning*

Minghai Chen†, Guiguang Ding†, Sicheng Zhao†, Hui Chen†, Jungong Han‡, Qiang Liu†
†School of Software, Tsinghua University, Beijing 100084, China
‡Northumbria University, Newcastle, NE1 8ST, UK
{Minghai.chen.12,schzhao,jichenhui2012}@gmail.com, {dinggg,liuqiang}@tsinghua.edu.cn, jungong.han@northumbria.ac.uk

Abstract

Image captioning is an important problem in artificial intelligence, related to both computer vision and natural language processing. There are two main problems in existing methods: in the training phase, it is difficult to find which parts of the captions are more essential to the image; in the caption generation phase, the objects or the scenes are sometimes misrecognized. In this paper, we consider the training images as the references and propose a Reference based Long Short Term Memory (R-LSTM) model, aiming to solve these two problems in one goal. When training the model, we assign different weights to different words, which enables the network to better learn the key information of the captions. When generating a caption, the consensus score is utilized to exploit the reference information of neighbor images, which might fix the misrecognition and make the descriptions more natural-sounding. The proposed R-LSTM model outperforms the state-of-the-art approaches on the benchmark dataset MS COCO and obtains top 2 position on 11 of the 14 metrics on the online test server.

Introduction

Benefiting from the advances of image classification and object detection, it becomes possible to automatically generate a sentence description for an image. This problem, known as image captioning, is of great importance to the goal of enabling computers to understand images. Besides recognizing the objects in the image (Lin et al. 2014), the generator should also be able to analyze their states, understand the relationship among them and express the information in natural language. Therefore, image captioning is a more challenging task involved in both computer vision and natural language processing, which can be exploited in cross-view retrieval (Ding et al. 2016; Lin et al. 2016).

The early efforts on image captioning mainly focused on organizing the recognized elements into sentences. These approaches used either templates (Farhadi et al. 2010; Yang et al. 2011; Kulkarni et al. 2011; Li et al. 2011) or pre-defined language models (Kuznetsova et al. 2012; Mitchell et al. 2012; Elliott and Keller 2013; Kuznetsova et al. 2014) in sentence generation, which normally end up with rigid and limited descriptions. Devlin et al. (2015b) simply used Nearest Neighbor to retrieve a description from the corpus for a given image, which reveals that nearest neighbors can provide valuable information.

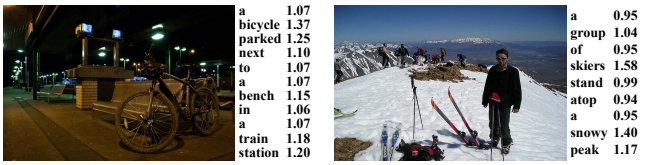


Figure 1: Word weighting examples. On the right of each image are the words of related captions in order and the weights assigned by our method.

Inspired by machine translation (Schwenk 2012; Cho et al. 2014), recent works employed recurrent neural network (RNN), especially long short term memory (LSTM) (Hochreiter and Schmidhuber 1997), to generate captions (Mao et al. 2014; Karpathy and Li 2015; Vinyals et al. 2015; Donahue et al. 2015; Fang et al. 2015), with the objective to maximize the likelihood of a sentence given the visual features of an image. In order to attend to salient visual concepts dynamically, different attention mechanisms are proposed (Jin et al. 2015; Xu et al. 2015; You et al. 2016). Despite achieving state-of-the-art performances, these methods treat different words of a caption in the same way, which makes it difficult to distinguish the important parts of the caption. Furthermore, the generated captions may be disturbed by unnecessary text content.

Obviously, in an image description, the words are not equally important. Take the first image of Figure 1 as an example, the word "bicycle" should be the most important since it defines the main subject of the image; "parked" is the status of the main subject and "bench" "train" "station" show the scene of the image, which should be less important; "next" "to" "a" "in" are relatively uninformative.

Motivated by these observations, we propose to make use of the labeled captions and the visual features of the training images as references to improve the generation quality.

*This research was supported by the National Natural Science Foundation of China (Grant No. 61571269) and the Royal Society Newton Mobility Grant (IE150997). Corresponding author: Guiguang Ding.

The references are incorporated in both the training phase and the generation phase of the LSTM model. In the training phase, the words in a caption are endowed with different weights according to their relevance to the corresponding image. A word with a higher relevance score indicates higher importance to describe the image, and thus a larger weight value is assigned to it when calculating the loss. In this way, the model could learn more in-depth information of the caption, such as what the principal objects are, which attributes are important to them and how they relate to each other. In the generation phase, we consider the nearest neighbors of the input image as references by combining the consensus score (Devlin et al. 2015a) and the likelihood of the generating sentence. The information provided by the nearest neighbors could help fix the misrecognition from the beginning, and better match the habit of human cognition. We also adjust the weight of the consensus score.

We evaluate the proposed method on the benchmark dataset MS COCO and the results demonstrate the significant superiority over the state-of-the-art approaches. We also report the performance of our method on the MS COCO Image Captioning Challenge. Comparing with all the latest approaches, we obtain the first place and the second place on both 5 of the total 14 metrics.

Related Work

Generally, the existing image captioning algorithms can be divided into three categories. The first category uses templates or designs a language model, which fill in slots of a template using co-occurrence relations gained from the corpus (Farhadi et al. 2010), conditional random field (Kulkarni et al. 2011), or web-scale n-gram data (Li et al. 2011). More complicated models have also been used to generate relatively flexible sentences. Mitchell et al. (2012) exploited syntactic trees to create a data-driven model. Elliott and Keller (2013) proposed visual dependency representation to extract relationships among the objects. However, all these models are heavily hand-designed or unexpressive.

The second category is based on the retrieval approaches. Some approaches (Gong et al. 2014; Hodosh, Young, and Hockenmaier 2013) took the input image as a query and selected a description in a joint image-sentence embedding space. Kuznetsova et al. (2012; 2014) retrieved images that are similar to the input image, extracted segments from their captions, and organized these segments into a sentence. Devlin et al. (2015b) simply found similar images and calculated the consensus score (Devlin et al. 2015a) of the corresponding captions to select the one with highest score. Usually retrieval based methods are unable to generate novel phrases or sentences, and thus are limited in image captioning. Notwithstanding, they indicate that we can take advantage of the images similar to the input image. This idea can be applied in other approaches, such as re-ranking candidate descriptions generated by other models (Mao et al. 2015). We also undertake this idea in our generation process.

New achievements come from the recent advantages in machine translation (Schwenk 2012; Cho et al. 2014), with the use of RNN. Mao et al. (2014) proposed a multimodal layer to connect a deep convolutional neural network (CNN) for images and a deep RNN for sentences, allowing the model to generate the next word given the input word and the image. Inspired by the encoder-decoder model (Cho et al. 2014) in machine translation, Vinyals et al. (2015) used a deep CNN to encode the image instead of a RNN for sentences, and then used LSTM (Hochreiter and Schmidhuber 1997), a more powerful RNN, to decode the image vector to a sentence. Many works follow this idea, and apply attention mechanisms in the encoder. Xu et al. (2015) extracted features from a convolutional layer rather than the fully connected layer. With each feature representing a fixed-size region of the image, the model can learn to change the focusing locations. Jin et al. (2015) employed a pre-trained CNN for object detection to analyze the hierarchically segmented image, and then ran attention-based decoder on these visual elements. Combining the whole image feature with the words obtained from the image by attribute detectors can also drive the attention model (You et al. 2016).

Similarly, our work follows the encoder-decoder model. But different from (Vinyals et al. 2015), the words in a caption are weighted in the training phase according to their relevance to the corresponding image, which well balances the model with the importance of a word to the caption. In the generation phase, we take advantage of the consensus score (Devlin et al. 2015a) to improve the quality of the sentences. Different from Mao et al. (2015) who simply used the consensus score to re-rank the final candidate descriptions, we use this score in the whole generation process, which means that the decoder takes the neighbors' information of the input image into account. With combination of the likelihood of a sentence, we propose a better evaluation function than just maximizing the likelihood.

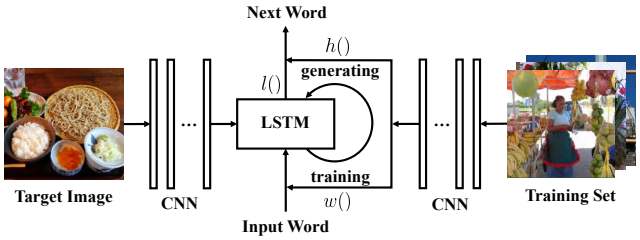


Figure 2: Overview of the proposed R-LSTM model. It is an encoder-decoder model (the left part) combined with the reference information extracted from the training set (the right part). The functions $w()$ and $h()$ indicate that the reference information is used to weight the input word when training and improve the output sentence when generating, respectively. $l()$ is the log likelihood.

Approach

The overview of the proposed image captioning method is shown in Figure 2. First, the deep VGG-16 model is employed as the encoder to extract CNN features of the target image and the training images. The weight attached to each word in the training captions is also calculated. Second, the LSTM model is trained using the weighted words and CNN features of the training images, and is adopted as decoder,

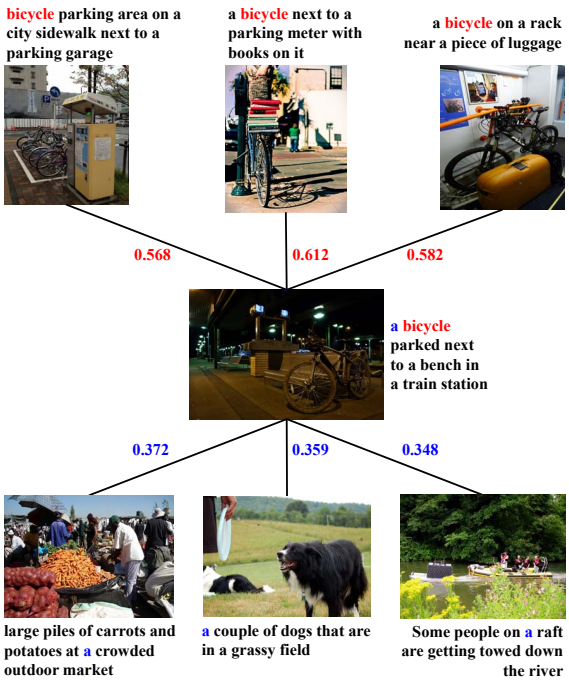


Figure 3: The K_σ values of the target image with some other images, whose captions contain "bicycle" or "a" respectively. It is obvious that the former images have higher K_σ values than the latter ones, suggesting that images labeled with "bicycle" are similar to the target image whose main subject is a bicycle, while the uninformative "a" leads to less similarity.

which takes the CNN features of the target image as input and generates the description words one by one. In this process, we jointly consider the likelihood and the consensus score as the evaluation function in beam search.

Weighted Training

Suppose I is a training image (also denote its encoded CNN features), $S = \{s_0, s_1, s_2, ..., s_N, s_{N+1}\}$ is the corresponding description sentence, where $\{s_1, s_2, ..., s_N\}$ is the original labeled words, s_0 is a special start word and s_{N+1} is a special stop word. Note that N depends on I. At time t, the likelihood of word s_t is decided by the input image I and previous words $s_0, s_1, ..., s_{t-1}$:

$$p(s_t|I, s_0, s_1, ..., s_{t-1}). \tag{1}$$

The joint log likelihood of description S is calculated by:

$$\log p(S|I) = \sum_{t=1}^{N+1} \log p(s_t|I, s_0, s_1, ..., s_{t-1}). \tag{2}$$

In the training phase, we take into consideration the words' importance by assigning different weights to the words, which aims to enable the model to concentrate on the main information of the captions. Higher weight will be given to the words indicating important elements such as the main subject, its status, the environment, *etc*. Suppose the

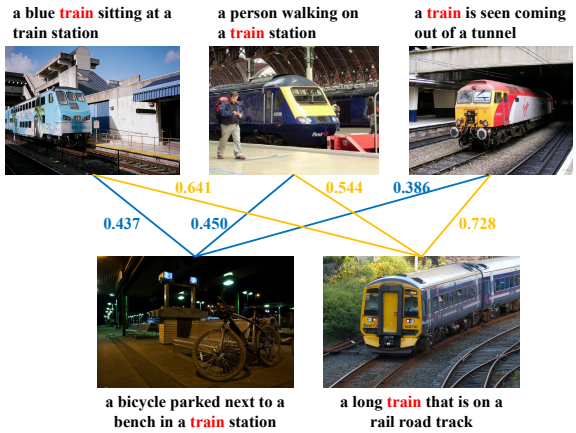


Figure 4: The K_σ values of two target images with some other images whose captions contain "train". For the first target image, "train" along with "station" denotes the scene of the image, while in the second target image "train" is the main subject. Therefore, the set of images containing "train" are more similar to the second target image, resulting in higher K_σ values.

weight of word s_t to image I is $w(s_t, I)$, then the model is trained to maximize the weighted log likelihood:

$$f(S, I) = \sum_{t=1}^{N+1} w(s_t, I) \log p(s_t|I, s_0, s_1, ..., s_{t-1}). \tag{3}$$

Note that in the training phase, the words $s_0, s_1, ..., s_t$ are given by the labeled caption. So their weights could be calculated as a preprocessing step.

Following the tag ranking approach (Liu et al. 2009), we calculate the weight of word s_i to image I as:

$$w(s_i, I) = \frac{\beta p(s_i|I)}{p(s_i)},\ i = 1, 2, \cdots, N, \tag{4}$$

where β is a parameter to ensure the average of all the weights is 1, and $p(s_i|I)$ denotes the likelihood of s_i in the captions of image I. The reason for dividing $p(s_i|I)$ by $p(s_i)$ is that a frequent word, such as "a" and "the", is not informative although it may appear in most descriptions.

Based on Bayes rule, we have

$$w(s_i, I) = \frac{\beta P(I|s_i) P(s_i)}{P(I) P(s_i)} = \frac{\beta P(I|s_i)}{P(I)}. \tag{5}$$

Since $P(I)$ is determined given image I, we can redefine Eq. (5) as:

$$w(s_i, I) \doteq \beta P(I|s_i). \tag{6}$$

Based on kernel density estimation (KDE) (Parzen 1962),

$$w(s_i, I) = \beta P(I|s_i) = \frac{\beta}{|G_{s_i}|} \sum_{I_j \in G_{s_i}} K_\sigma(I - I_j), \tag{7}$$

where G_{s_i} denotes the set of images whose captions contain word s_i, and the Gaussian kernel function K_σ is defined as:

$$K_\sigma(I - I_j) = \exp(-\frac{(\|I - I_j\|)^2}{\sigma^2}), \tag{8}$$

Table 1: Performance (%) of the weighting method on MS COCO dataset, where "-ft" refers to finetuning the CNN encoder.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Original	65.0	46.2	32.7	23.5	20.9	47.7	70.2
Weighted	65.1	46.7	33.4	24.2	21.3	47.9	71.8
Original-ft	71.3	54.0	40.2	30.1	24.7	52.6	92.8
Weighted-ft	**71.7**	**54.6**	**40.7**	**30.4**	**24.8**	**52.7**	**94.0**

where the adius parameter σ is set as the the average distance of each two images in the training set, and the image vectors are extracted from a deep CNN. Therefore, in a set of images containing a same description word, if an image is very similar to others, it is natural to infer that the word is very relevant to the image and thus will be assigned with a high weight in the image's captions; Otherwise, if an image does not look like other images, which means that the word is not important or is even noise to the image, the word will be given a low weight. Eq. (8) is meaningful in two aspects: it measures the importance of different words in a same caption (Figure 3) and the importance of a word to different images (Figure 4).

Generation Using Reference

After training, the model can generate a description $R = \{r_0, r_1, r_2, ..., r_M, r_{M+1}\}$ (r_0 and r_{M+1} are special start word and stop word respectively) given a target image J, with the objective to maximize:

$$g(R, J) = (1 - \alpha)l(R, J) + \alpha h(R, J), \tag{9}$$

where $h(R, J)$ is the consensus score of sentence R, and $l(R, J)$ is the log likelihood:

$$l(R, J) = \log p(R|J) = \sum_{t=1}^{M+1} \log p(r_t|J, r_0, r_1, ..., r_{t-1}). \tag{10}$$

The consensus score comes from the idea that the descriptions of similar images are very helpful in image captioning. Some retrieval-based methods directly use the captions of similar images as the description of the input image. Devlin et al. (2015b) used a simple k-Nearest Neighbor model. First, retrieve k nearest neighbors of the input image and get the set of their captions $C = \{c_1, c_2, ..., c_{5k}\}$ (5 captions for each image). Second, calculate the n-gram overlap F-score for every two captions in C. The consensus score of c_i is defined as the mean of its top m F-scores. Finally, select the caption with the highest consensus score as the description of the input image.

Similar to (Devlin et al. 2015b), we calculate the consensus score $h(R, J)$ for image J and the generated sentence R (including incomplete ones that are being generated by the decoder) as:

$$h(R, J) = \frac{1}{|C_J|} \sum_{c \in C_J} sim(R, c), \tag{11}$$

where C_J is the caption set of the k-Nearest Neighbor images of image J, and $sim(\cdot, \cdot)$ is the function to calculate the similarity between two sentences (we use BLEU-4 (Papineni et al. 2002) in experiments).

Since $l(R, J)$ is much larger than $h(R, J)$ in terms of absolute value, we normalize them before linear weighting:

$$\begin{aligned} l'(R, J) &= \frac{l(R, J) - \min_{c \in H} l(c, J)}{\max_{c \in H} l(c, J) - \min_{c \in H} l(c, J)}, \\ h'(R, J) &= \frac{h(R, J) - \min_{c \in H} h(c, J)}{\max_{c \in H} h(c, I) - \min_{c \in H} h(c, J)}, \end{aligned} \tag{12}$$

where H is the set of generated candidate descriptions. Now we get the final evaluation function:

$$g(R, J) = (1 - \alpha)l'(R, J) + \alpha h'(R, J),\ 0 \leq \alpha \leq 1. \tag{13}$$

Different from training, in the generation phase the input word at time t is the output word r_{t-1}, instead of the word in the labeled caption. Besides, as our dictionary size is large (On MS COCO dataset we obtain $\sim$10000 words after filtering out infrequent ones), the searching space is too large for exhaustive enumeration. Therefore, we implement the beam search as an approximation. At each time step, we keep a set of K (called "beam size") best sentences from K^2 candidates according to Eq. (13). When a sentence is completed (the next word generated by the decoder is the stop word, or the sentence reaches the maximum length), it will be moved to the final pool, which also has the size of K and is maintained according to Eq. (13).

Experiments

To evaluate the effectiveness of the proposed method, we carry out experiments on the popular MS COCO dataset, which contains 123,287 images labeled with at least 5 captions by different AMT workers. Since there is no standardized split on MS COCO, we use the public available split[1] as in previous works ((Karpathy and Li 2015; Xu et al. 2015; You et al. 2016), *etc.*). Following the evaluation API provided by the MS COCO server, we report the results on different metrics, including BLEU-1, 2, 3, 4, METEOR, ROUGE-L and CIDEr. Similar to (Jin et al. 2015), the beam size K used in the beam search is set to 10.

Results on Weighted Word Training

Some of the weighted words are shown in Figure 1. Take the second image for example, the weight of the subject "skiers" is the largest, followed by the background "snowy" and "peak". As "snowy" is more obvious than "peak", its weight is relatively larger. We can conclude that after weighting, the subjects in the image are emphasized.

The performance of the LSTM networks trained before and after weighting the words is shown in Table 1. We can see that the performance is improved by weighting, no matter finetuning the CNN encoder or not.

[1] https://github.com/karpathy/neuraltalk

Table 2: Performance (%) of the proposed model compared with several state-of-the-art methods on MS COCO dataset.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Google NIC (Vinyals et al. 2015)	66.6	45.1	30.4	20.3	-	-	-
Toronto (Xu et al. 2015)	71.8	50.4	35.7	25.0	23.0	-	-
ATT (You et al. 2016)	70.9	53.7	40.2	30.4	24.3	-	-
USC (Jin et al. 2015)	69.7	51.9	38.1	28.2	23.5	50.9	83.8
m-RNN (Mao et al. 2015)	71.4	54.3	40.6	30.4	23.9	51.9	93.8
LRCN (Donahue et al. 2015)	71.4	54.3	40.2	29.7	24.2	52.4	88.9
R-LSTM (ours)	**76.1**	**59.6**	**45.0**	**33.7**	**25.7**	**55.0**	**102.9**

Table 3: Evaluation results (%) of the latest captioning methods on dataset c5 and c40 on the online MS COCO server (http://mscoco.org/dataset/#captions-leaderboard). The subscripts indicate the ranking of the individual algorithms with respect to the corresponding metrics on July 12, 2016.

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
MSM@MSRA	73.9_2	$\mathbf{91.9_1}$	57.5_2	$\mathbf{84.2_1}$	43.6_2	$\mathbf{74.0_1}$	$\mathbf{33.0_1}$	$\mathbf{63.2_1}$	$\mathbf{25.6_1}$	$\mathbf{35.0_1}$	$\mathbf{54.2_1}$	$\mathbf{70.0_1}$	$\mathbf{98.4_1}$	$\mathbf{100.3_1}$
THU-MIG (ours)	$\mathbf{75.1_1}$	91.3_2	$\mathbf{58.3_1}$	83.3_2	$\mathbf{43.6_1}$	72.7_2	32.3_2	61.6_3	25.1_5	33.6_6	54.1_2	68.8_2	96.9_2	98.8_2
AugmentCNNwithDet	72.1_8	90.5_5	55.3_8	81.5_6	41.6_7	70.6_7	31.5_5	59.7_7	25.1_6	34.0_4	53.1_6	68.3_4	95.6_3	96.8_4
ChalLS	72.3_6	89.8_9	55.3_9	80.9_8	41.4_9	70.1_9	30.9_8	59.0_9	25.2_3	34.0_3	53.1_5	67.9_9	95.5_4	97.0_3
ATT (You et al. 2016)	73.1_{45}	90.0_8	56.5_4	81.5_7	42.4_3	70.9_6	31.6_4	59.9_6	25.0_7	33.5_8	53.5_3	68.2_6	94.3_5	95.8_5

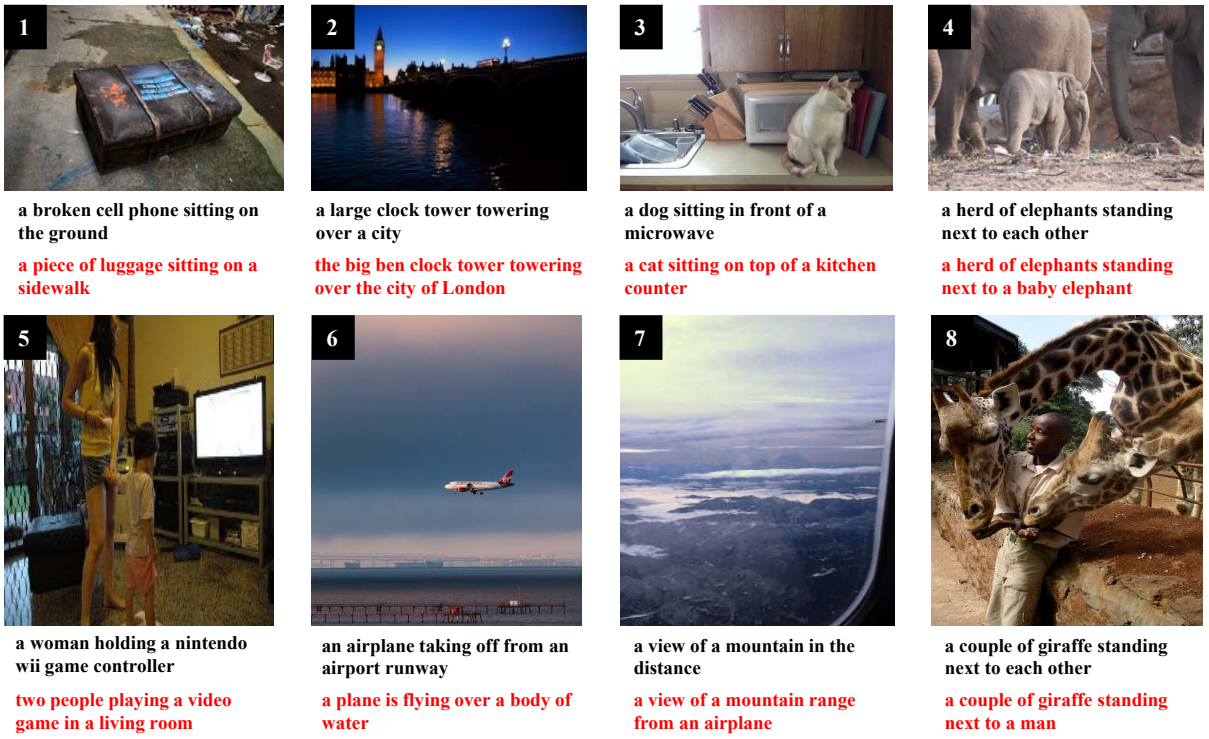


Figure 5: Examples of generated captions by our model before (in black) and after (in red) the use of references.

Results on Reference Based Generation

The parameter α in Eq. (13) is crucial in our methods, which determines to what extent the generator depends on references. The black line in Figure 6(a) shows how the quality of generated captions (on CIDEr) varies with respect to α. With the increase of α, the performance firstly becomes better and then turns worse, which demonstrates that referring neighbor images can improve the performance and that relying too much on references also leads to poor performance.

In the generation phase, the sentence length is increasing. Since a sentence contains more information when it has more words, it may not be a good idea to keep the same weight of the references. We try to change α in different generation stages. For example, in the early stage we set $\alpha = \alpha_1$, and in the final pool stage we set $\alpha = \alpha_2$. We

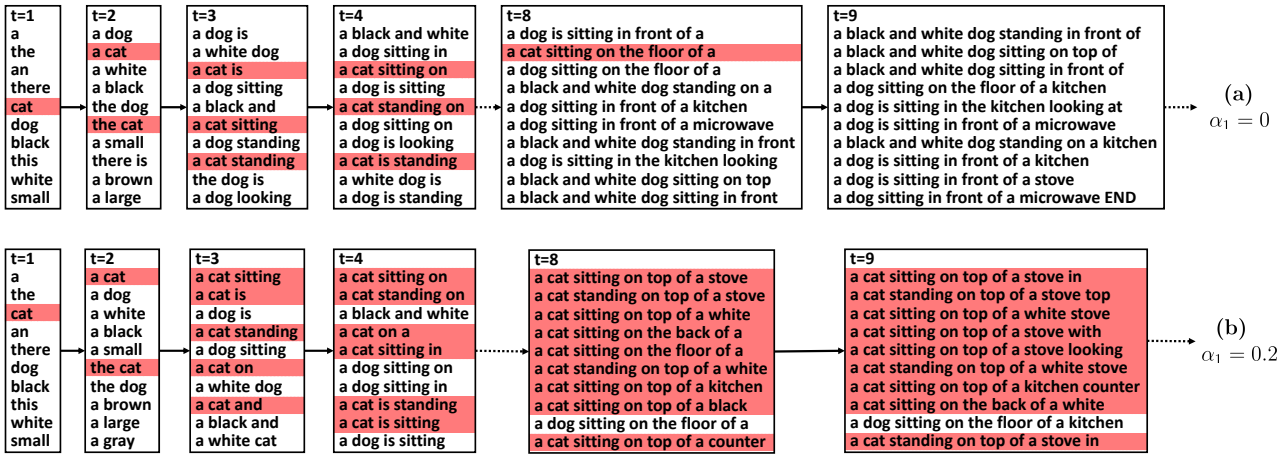
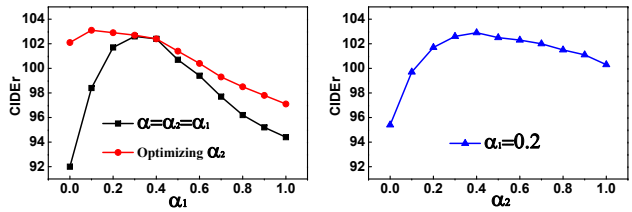


Figure 7: The beam search process of the 3rd image in Figure 5 ranked by Eq. (13) when (a) $\alpha_1 = 0$, (b) $\alpha_1 = 0.2$. The red lines are the generating captions correctly recognizing the subject "cat".



(a) (b)

Figure 6: (a) The black and red lines are the influence of parameter α (i.e. $\alpha_2 = \alpha_1 = \alpha$) and α_1 (when optimizing α_2) in the proposed generator, respectively. (b) Performance of the generator with different α_2 when $\alpha_1 = 0.2$.

conduct experiment with $\alpha_1 = 0.2$ fixed and adjust α_2. As shown in Figure 6(b), we can obtain better performance by varying α_2 from 0.2 to 0.4. We repeat this process for $\alpha_1 = 0, 0.1, 0.2, ..., 1$, and they all perform better by adjusting α_2 (the red line in Figure 6(a)). We report the results when $\alpha_1 = 0.2$, $\alpha_2 = 0.4$ in the following experiments unless otherwise specified.

The comparison of the completed model and several state-of-the-art methods is shown in Table 2, where "-" represents unknown scores. It is clear that our approach performs the best on all the metrics, respectively achieving 4.3%, 5.3%, 4.4%, 3.3%, 1.4%, 2.6% and 9.1% improvement compared with previous best results. We also test our approach on the online MS COCO server. The results compared with the latest methods are reported in Table 3. Despite keen competition, we obtain the first and second places on 3 and 8 metrics respectively, at the time of submission.

Case Study

Some examples of the generated sentences are illustrated in Figure 5. The captions in red show how the use of references improves the generation quality: misrecognizion is fixed in images 1, 3, 6; more semantic details are given in image 2 (the model infers that the tower is the Big Ben and the city is London), image 4 (a baby elephant), image 7 (from an airplane) and image 8 (there is a man next to the giraffes); better match the habit of human cognition in image 3 (on top of a kitchen counter v.s. in front of a microwave) and image 5 (when holding a game controller, the people are actually playing a video game).

We take the 3rd image in Figure 5 for example to understand the beam search process of Eq. (13), i.e. the significance of $\alpha_1 \neq 0$. We can see that the subject "cat" is misrecognized as "dog" without using the consensus score, whose beam search process is illustrated in Figure 7 (a). At the beginning the model is waving between "dog" and 'cat", more possibly to be "dog". As $\alpha_1 = 0$, the model cannot utilize the neighbor images to correct the mistake. When $t = 9$, there is no "cat" in the candidate sentences. No matter how much is α_2, this mistake cannot be corrected. However, when $\alpha_1 \neq 0$, this situation can be avoided with the help of references, as shown in Figure 7 (b).

Conclusion

In this paper, we proposed a reference based LSTM model, making use of training images as references to improve the quality of generated captions. In the training phase, the words are weighted according to their relevance to the image, which enables the model to focus on the key information of the captions. In the generation phase, a novel evaluation function is proposed by combining the likelihood with the consensus score, which could fix misrecognition and make the generated sentences more natural-sounding. Experimental results on MS COCO corroborated that the proposed R-LSTM is superior over the state-of-the-art approaches for image captioning. In further studies, we plan to add the attention mechanisms into the reference model and try other weighting strategies. How to generate image captions with emotion (Zhao et al. 2014; Zhao et al. 2017b) and sentiment (Mathews, Xie, and He 2016) and extend it to personalized settings (Zhao et al. 2017a) is also worth studying.

References

[Cho et al. 2014] Cho, K.; Van Merriënboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 1724–1734.

[Devlin et al. 2015a] Devlin, J.; Cheng, H.; Fang, H.; Gupta, S.; Deng, L.; He, X.; Zweig, G.; and Mitchell, M. 2015a. Language models for image captioning: The quirks and what works. In *ACL*, 100–105.

[Devlin et al. 2015b] Devlin, J.; Gupta, S.; Girshick, R.; Mitchell, M.; and Zitnick, C. L. 2015b. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*.

[Ding et al. 2016] Ding, G.; Guo, Y.; Zhou, J.; and Gao, Y. 2016. Large-scale cross-modality search via collective matrix factorization hashing. *IEEE Transactions on Image Processing* 25(11):5427–5440.

[Donahue et al. 2015] Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2625–2634.

[Elliott and Keller 2013] Elliott, D., and Keller, F. 2013. Image description using visual dependency representations. In *EMNLP*, 1292–1302.

[Fang et al. 2015] Fang, H.; Gupta, S.; Iandola, F.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; et al. 2015. From captions to visual concepts and back. In *CVPR*, 1473–1482.

[Farhadi et al. 2010] Farhadi, A.; Hejrati, M.; Sadeghi, M. A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; and Forsyth, D. 2010. Every picture tells a story: Generating sentences from images. In *ECCV*, 15–29.

[Gong et al. 2014] Gong, Y.; Wang, L.; Hodosh, M.; Hockenmaier, J.; and Lazebnik, S. 2014. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 529–545.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

[Hodosh, Young, and Hockenmaier 2013] Hodosh, M.; Young, P.; and Hockenmaier, J. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47:853–899.

[Jin et al. 2015] Jin, J.; Fu, K.; Cui, R.; Sha, F.; and Zhang, C. 2015. Aligning where to see and what to tell: image caption with region-based attention and scene factorization. *arXiv preprint arXiv:1506.06272*.

[Karpathy and Li 2015] Karpathy, A., and Li, F.-F. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 3128–3137.

[Kulkarni et al. 2011] Kulkarni, G.; Premraj, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A. C.; and Berg, T. L. 2011. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 1601–1608.

[Kuznetsova et al. 2012] Kuznetsova, P.; Ordonez, V.; Berg, A. C.; Berg, T. L.; and Choi, Y. 2012. Collective generation of natural image descriptions. In *ACL*, 359–368.

[Kuznetsova et al. 2014] Kuznetsova, P.; Ordonez, V.; Berg, T. L.; and Choi, Y. 2014. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics* 2(10):351–362.

[Li et al. 2011] Li, S.; Kulkarni, G.; Berg, T. L.; Berg, A. C.; and Choi, Y. 2011. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, 220–228.

[Lin et al. 2014] Lin, Z.; Ding, G.; Hu, M.; Lin, Y.; and Ge, S. S. 2014. Image tag completion via dual-view linear sparse reconstructions. *Computer Vision and Image Understanding* 124:42–60.

[Lin et al. 2016] Lin, Z.; Ding, G.; Han, J.; and Wang, J. 2016. Cross-view retrieval via probability-based semantics-preserving hashing. *IEEE Transactions on Cybernetics*.

[Liu et al. 2009] Liu, D.; Hua, X.-S.; Yang, L.; Wang, M.; and Zhang, H.-J. 2009. Tag ranking. In *WWW*, 351–360.

[Mao et al. 2014] Mao, J.; Xu, W.; Yang, Y.; Wang, J.; and Yuille, A. L. 2014. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*.

[Mao et al. 2015] Mao, J.; Xu, W.; Yang, Y.; Wang, J.; and Yuille, A. L. 2015. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR*.

[Mathews, Xie, and He 2016] Mathews, A. P.; Xie, L.; and He, X. 2016. Senticap: Generating image descriptions with sentiments. In *AAAI*, 3574–3580.

[Mitchell et al. 2012] Mitchell, M.; Han, X.; Dodge, J.; Mensch, A.; Goyal, A.; Berg, A.; Yamaguchi, K.; Berg, T.; Stratos, K.; and Daumé III, H. 2012. Midge: Generating image descriptions from computer vision detections. In *EACL*, 747–756.

[Papineni et al. 2002] Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

[Parzen 1962] Parzen, E. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics* 33(3):1065–1076.

[Schwenk 2012] Schwenk, H. 2012. Continuous space translation models for phrase-based statistical machine translation. In *COLING*, 1071–1080.

[Vinyals et al. 2015] Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*, 3156–3164.

[Xu et al. 2015] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2048–2057.

[Yang et al. 2011] Yang, Y.; Teo, C. L.; Daumé III, H.; and Aloimonos, Y. 2011. Corpus-guided sentence generation of natural images. In *EMNLP*, 444–454.

[You et al. 2016] You, Q.; Jin, H.; Wang, Z.; Fang, C.; and Luo, J. 2016. Image captioning with semantic attention. In *CVPR*, 4651–4659.

[Zhao et al. 2014] Zhao, S.; Gao, Y.; Jiang, X.; Yao, H.; Chua, T.-S.; and Sun, X. 2014. Exploring principles-of-art features for image emotion recognition. In *ACM MM*, 47–56.

[Zhao et al. 2017a] Zhao, S.; Yao, H.; Gao, Y.; Ding, G.; and Chua, T.-S. 2017a. Predicting personalized image emotion perceptions in social networks. *IEEE Transactions on Affective Computing*.

[Zhao et al. 2017b] Zhao, S.; Yao, H.; Gao, Y.; Ji, R.; and Ding, G. 2017b. Continuous probability distribution prediction of image emotions via multi-task shared sparse regression. *IEEE Transactions on Multimedia*.