MemCap: Memorizing Style Knowledge for Image Captioning

Wentian Zhao[1], **Xinxiao Wu**[1*], **Xiaoxun Zhang**[2]
[1] Lab. of IIT, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China
[2] Alibaba Group
{wentian_zhao, wuxinxiao}@bit.edu.cn, xiaoxun.zhang@alibaba-inc.com

Abstract

Generating stylized captions for images is a challenging task since it requires not only describing the content of the image accurately but also expressing the desired linguistic style appropriately. In this paper, we propose MemCap, a novel stylized image captioning method that explicitly encodes the knowledge about linguistic styles with memory mechanism. Rather than relying heavily on a language model to capture style factors in existing methods, our method resorts to memorizing stylized elements learned from training corpus. Particularly, we design a memory module that comprises a set of embedding vectors for encoding style-related phrases in training corpus. To acquire the style-related phrases, we develop a sentence decomposing algorithm that splits a stylized sentence into a style-related part that reflects the linguistic style and a content-related part that contains the visual content. When generating captions, our MemCap first extracts content-relevant style knowledge from the memory module via an attention mechanism and then incorporates the extracted knowledge into a language model. Extensive experiments on two stylized image captioning datasets (SentiCap and FlickrStyle10K) demonstrate the effectiveness of our method.

Introduction

The research on image captioning has made remarkable progress in recent years. Most existing image captioning models (Vinyals et al. 2015) (Karpathy and Fei-Fei 2015) (Anderson et al. 2018) focus on generating accurate descriptions, while ignoring the linguistic style of the sentences. Ideally, a practical image captioning model should not only describe the visual content accurately, but also be able to incorporate specific linguistic style into sentences appropriately. Such stylized image captioning model is particularly valuable in many scenarios, including generating attractive image or video titles for better recommendation, or automatically writing image descriptions that are interesting for children in early education.

For generating high-quality stylized captions conditioned on an input image, it is significantly important to effectively integrate factual image content and suitable style-related phrases. However, some style-related information can not be directly perceived from the image since such information is not visually grounded. Under such circumstances, human beings can still describe the image with desired styles, owing to the ability of association empowered by their prior knowledge. For instance, when someone is asked to write a positive sentence for the image shown in Figure. 1, he might express that the food is delicious according to the association between the positive expression "delicious" and the noun "food" in prior knowledge, although the actual taste of the food is not shown in the image. Inspired by this, we explore how to learn the knowledge about linguistic styles and how to utilize such knowledge for stylized image captioning in a reasonable way, imitating the language expressing procedure of human beings.

Factual: The plate has a sandwich with many large french fries.
Positive: A plate of delicious food including French fries.
Negative: A plate of disgusting food found at a diner.

Figure 1: An example of factual image description and stylized image description. The style related words are colored in red.

In this paper, we propose a MemCap method for stylized image captioning. It first memorizes the knowledge concerned with phrases that reflect the linguistic style, referred to as *style knowledge*, and then incorporates such knowledge into textual descriptions. To be specific, we design a *style memory module* to encode the style knowledge learned from the training corpus via a set of embedding vectors. However, it is difficult to learn the style knowledge since content-related phrases and style-related phrases usually co-exist in a stylized sentence. To separate the style-related phrases from training corpus, we develop an algorithm, referred to as *sentence decomposing algorithm*, to split a stylized sentence into style-related part and content-related part in an unsuper-

[*]Corresponding author: Xinxiao Wu

vised manner. Specifically, the algorithm performs sentence compression operation based on a dependency tree to preserve only the factual content in the stylized sentence, and the phrases that are removed from the dependency tree are identified as style-related phrases.

To generate stylized captions for an image, we apply an attention mechanism to extract the relevant knowledge from the style memory module by learning attention weights according to the image content. The extracted style knowledge is then integrated with the visual representation of the image as the input to a language model. Since our method is trained with unpaired stylized corpus, an intermediate form between images and factual sentences is indispensable. In this work, we use scene graph as the intermediate form that summarizes the objects, relationships between objects and attributes of objects in a visual scene. Both the images and the factual content of sentences are represented by scene graphs.

The main contributions of this paper are:

- We propose a MemCap method for stylized image captioning, where a style memory module is designed to explicitly memorize the style knowledge learned from large corpus.
- We propose a sentence decomposing algorithm that automatically separates style-related part from stylized sentence to facilitate the learning of the style memory module.
- Extensive experiments on several datasets demonstrate the superior performance of our method compared with the state-of-the-art methods.

Related Work

Stylized Image Captioning

Stylized image captioning has attracted growing attention recently. Mathews et al. (Mathews, Xie, and He 2016) first proposed the switching RNN which can generate image descriptions with positive or negative sentiments. Chen et al. (Chen et al. 2018) proposed style-factual LSTM and an adversarial training approach to train the stylized image captioning model. All these methods depend heavily on stylized sentences with paired images for training a stylized image captioning model.

To reduce the dependency on paired data, several recent methods (Gan et al. 2017) (Mathews, Xie, and He 2018) (Chen et al. 2019) (Guo et al. 2019) have been proposed to leverage unpaired stylized corpus. Gan et al. proposed the StyleNet model (Gan et al. 2017) that decomposes the weight matrices in the Long-Short Term Memory (LSTM) network to model both factual sentences and stylized sentences. Sinn et al. (Shin, Ushiku, and Harada 2016) proposed to incorporate sentiment terms into image descriptions with the aid of an additional CNN. Chen et al. (Chen et al. 2019) proposed to generate stylized image descriptions with domain layer norm, which enables generating various stylized descriptions. MSCap (Guo et al. 2019) is proposed to generate image descriptions in multiple styles by training a single captioning model on unpaired stylized corpus, with the help of several auxiliary modules. All these methods focus on designing language models or training algorithms to capture style factors for generating stylized captions. In contrast, our method resorts to explicitly encode the style knowledge learned from large corpus by building a memory module.

Text Style Transfer

The studies of text style transfer are also closely related to our work. Early methods (Jhamtani et al. 2017) apply supervised learning to train a sequence-to-sequence model that can modify the linguistic style of the input text. However, this requires a large amount of paired training corpus, which is difficult to obtain. Recent text style transfer methods focus on utilizing large-scale unpaired corpus. Shen et al. (Shen et al. 2017) propose to align the style-irrelevant representation of sentences in different domains, aiming at preserving the content of the sentence. Some methods (Prabhumoye et al. 2018) (Xu et al. 2018) enforce content preservation by applying back-translation mechanism. Zhang et al. (Zhang et al. 2018) propose to learn sentiment memories for text style transfer. Compared to (Zhang et al. 2018), the task of stylized image captioning has to deal with the larger gap between images and natural language. In addition, our model is capable of generating sentences in more complex linguistic styles, including positive, negative, humorous and romantic. Furthermore, our model is optimized using self-critical training with carefully designed reward.

Our Method

Overview

Given an input image x and a style label s, a stylized image captioning model is expected to generate a sentence $\hat{y}^s$ that preserves the content in the image x with the style s. To train the stylized image captioning model, we are given paired factual data $D_f = \{(x_i, y_i^f)|_i\}$, where x_i and y_i^f denote the i-th image and its corresponding factual description, respectively, and large scale unpaired stylized sentences with K different styles. The corpus of each style is denoted as $D_s = \{y_i^s|_i\}$, where y_i^s represents the i-th stylized sentence in D_s and $s \in \{s_1, s_2, ..., s_K\}$ represents the style label.

The overview of our model is illustrated in Figure 2. Our proposed model consists of a style memory module $\mathcal{M}$, a sentence decomposer $\mathcal{P}$, a captioner $\mathcal{C}$, an image scene graph generator $\mathcal{E}$ and a sentence scene graph generator $\mathcal{F}$. During testing, for each image input x, a scene graph G^x is generated by the image scene graph generator $\mathcal{E}$ to summarize the content of x, denoted as $G^x = \mathcal{E}(x)$. Then the content-relevant style knowledge $\boldsymbol{m}$ is extracted from the style memory module $\mathcal{M}$ according to G^x, denoted as $\boldsymbol{m} = \mathcal{M}(G^x)$. Finally, the stylized sentence $\hat{y}^s$ is generated by the captioner $\mathcal{C}$ with G^x and $\boldsymbol{m}$, denoted as $\hat{y}^s = \mathcal{C}(G^x, \boldsymbol{m})$.

During the training of style memory module $\mathcal{M}$, we split each stylized training sentence y^s into a content-related part W_c and a style-related part W_s by the sentence decomposer $\mathcal{P}$. The content-related part W_c is then fed into the sentence scene graph generator $\mathcal{F}$ to generate its scene graph G^y, i.e., $G^y = \mathcal{F}(W_c)$. The style-related part W_s is used to update

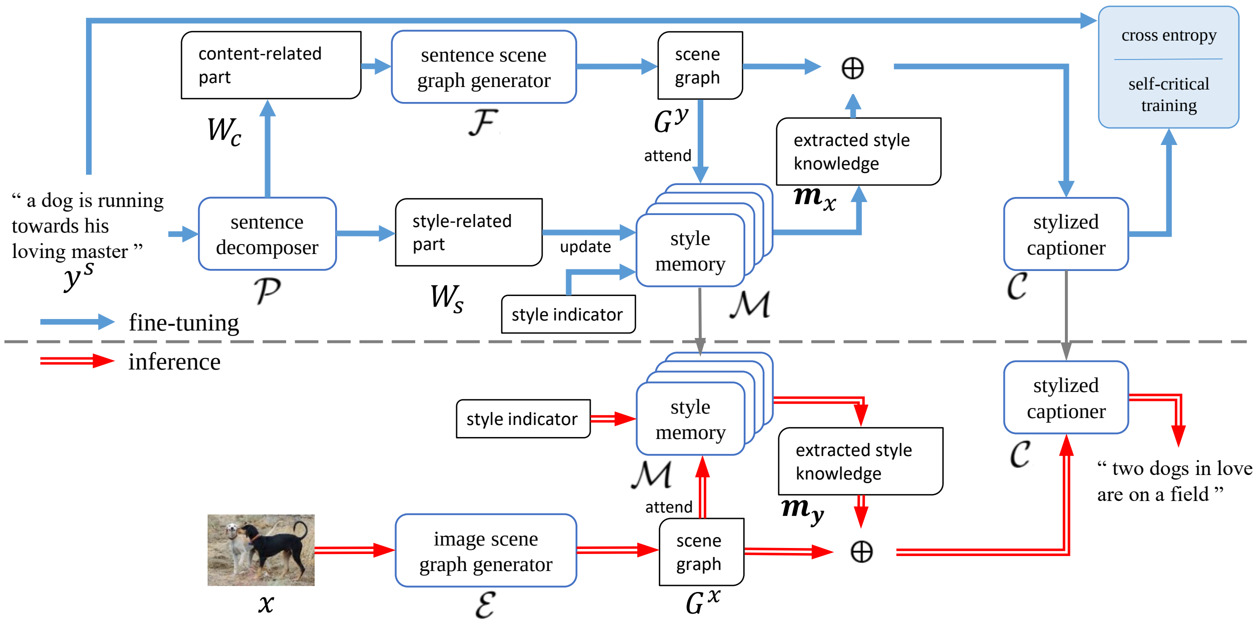

Figure 2: Overview of our proposed method. The blue arrows indicate the training process using unpaired stylized sentences and the red arrows indicate the inference process. During training, each stylized sentence y^s is split into a content-related part W_c that is encoded as scene graph G^y, and a style-related part W_s that is used to update the memory module $\mathcal{M}$. The style knowledge $\boldsymbol{m}_y$ is then extracted according to the scene graph G^y, and is input into the captioner $\mathcal{C}$ together with G^y. During inference, an image x is encoded in a scene graph G^x, and the style knowledge $\boldsymbol{m}_x$ is extracted according to G^x. Similar to training process, G^x and $\boldsymbol{m}_x$ are fed into captioner $\mathcal{C}$ to generate stylized caption.

the style memory module $\mathcal{M}$ by weighing and adding the embedding of W_s to each column in $\mathcal{M}$.

During the training of the captioner $\mathcal{C}$, since only the training sentences are available, the scene graph G^y derived from the content-related part of sentence y^s is used as one input to $\mathcal{C}$ instead of G^x. The style knowledge is also extracted by G^x. We compare the sentence $\hat{y}^s = \mathcal{C}(G^y, \mathcal{M}(G^y))$ with the training sentence y^s to optimize the captioner.

The style memory module $\mathcal{M}$ and captioner $\mathcal{C}$ are trained in an end-to-end manner. A traditional cross-entropy loss function and a self-critical training strategy (Rennie et al. 2017) with the reward function designed for stylized image captioning are successively utilized to optimize $\mathcal{M}$ and $\mathcal{C}$.

Stylized Sentence Decomposing

The sentence decomposer $\mathcal{P}$ is implemented by an iterative sentence decomposing that separates the style-related part W_s and content-related part W_c of a sentence. W_s contains phrases that reflect the linguistic style of the sentence and W_c contains the scenes, objects and actions that the sentence describes. Formally, given a sentence $y = w_1, w_2, ..., w_L$, this algorithm assigns a label $l_i \in \{0, 1\}$ for each word w_i, indicating whether w_i is style-related. For a factual sentence, the style-related part is an empty sequence.

Since the style-related phrases rarely appear in factual sentences, a stylized sentence leads to a higher perplexity than a factual sentence when being evaluated by a language model trained with factual sentences. Thus, we train a language model with factual sentences as a guidance to distinguish between the content-related part and the style-related part. For a stylized sentence y^s, we parse the sentence with a dependency tree parser, and then prune the dependency tree to preserve the content-related part. Each word w_i in the sentence corresponds to a node v_i in the dependency tree. A directed edge e_{ij} from v_i to v_j indicates that word w_j is dependent on word w_i. In the t-th iteration, we enumerate all the edges in the dependency tree of sentence $y^s_{(t-1)}$. For edge e_{ij}, we attempt to remove the node v_j and its subtree and the remaining nodes form a new sentence $\hat{y}^s_{(t,j)}$. All the new sentences in the t-th iteration $\hat{y}^s_{(t,*)}$ are evaluated by the language model pre-trained on factual sentences. The sentence with the lowest perplexity is saved for the next iteration. If the perplexity of all the new sentences are higher than the perplexity of original sentence, the whole pruning process ends. The words in the last pruned sentence $y^s_{(t)}$ comprises the content-related part of the sentence, which are assigned with label $l_i = 0$. The words in the pruned part are regarded as style-related part, which are assigned with label $l_i = 1$.

Scene Graph Generation

In this section, we illustrate the details of the image scene graph generator $\mathcal{E}$ and the sentence scene graph generator $\mathcal{F}$, respectively. The scene graph summarizes the information in an image or a sentence in a structured form, including the objects, the relationship between objects and the attributes of objects in the image or the sentence. A scene graph G is comprised of a node set V, and an edge set E, denoted as $G = (V, E)$. The node set is comprised of three different kinds of nodes, including object, relationship and attribute. We denote the i-th object as o_i, the relationship between two objects o_i and o_j as r_{ij} and the k-th attribute of an object o_i as a_i^k. An edge from o_i to r_{ij} and another edge from r_{ij} to o_j can be represented by a triplet $\langle o_i, r_{ij}, o_j\rangle$, where o_i, r_{ij} and o_j correspond to the subject, the predicate and the object in the triplet.

We use the method in (Anderson et al. 2016) to convert a sentence into a scene graph, which involves two stages. The sentence is first converted to a dependency tree using a dependency parser (Klein and Manning 2003). A rule-based method (Schuster et al. 2015) is then applied to map the dependency tree to a scene graph. For a stylized sentence y^s, we decompose the sentence and generate the sentence scene graph using the content-related part W_c rather than the whole sentence.

To generate the scene graph of an image, we first generate the factual description of the image and then convert the sentence to the scene graph. Specifically, we train the Up-Down captioning model proposed in (Anderson et al. 2018) to generate the factual description for image x, which is then converted to scene graph G^x using the aforementioned method.

Style Memory Module

Definition After separating the style-related part and content-related part of the training sentences, we use a style memory module to encode the style-related words or phrases during training. The embedding vectors that contain knowledge with regard to style s forms a matrix $\boldsymbol{M}_s \in \mathbb{R}^{d\times p}$, where p is the number of the embedding vectors. An additional matrix $\boldsymbol{M}_s' \in \mathbb{R}^{d\times p}$ stores the factual content corresponding to the style knowledge in $\boldsymbol{M}_s$.

Memory Update Given a stylized sentence $y^s = [w_1, w_2, ..., w_L]$ with the corresponding style labels of words $\{l_1, l_2, ..., l_L\}$, the style memory is updated with the embeddings of style-related words. The base vectors in $\boldsymbol{M}_s$ and $\boldsymbol{M}_s'$ are attended with the embeddings of style-related words in the sentence, and are both updated according to the attention weights. Inspired by (Zhang et al. 2018), the update operation is formulated as

$$\begin{aligned}
\boldsymbol{e}_s &= \sum_{i=0}^{L} l_i \boldsymbol{e}_{w_i}, \\
\hat{\boldsymbol{\alpha}} &= (\boldsymbol{M}_s')^\top \boldsymbol{e}_c, \\
\boldsymbol{\alpha} &= \mathrm{softmax}(\hat{\boldsymbol{\alpha}}), \\
\boldsymbol{M}_s' &= \boldsymbol{M}_s' + \boldsymbol{e}_c \boldsymbol{\alpha}^\top, \\
\boldsymbol{M}_s &= \boldsymbol{M}_s + \boldsymbol{e}_s \boldsymbol{\alpha}^\top,
\end{aligned} \tag{1}$$

where $\boldsymbol{e}_{w_i} \in \mathbb{R}^d$ denotes the d-dimensional embedding vector of word w_i, $\boldsymbol{e}_s$ denotes the embedding of style-related words in y^s, $\boldsymbol{e}_c$ denotes the embedding of scene graph, which will be further explained in the next section. The vector $\boldsymbol{\alpha} \in \mathbb{R}^{1\times p}$ denotes the weights for each embedding vector in the memory module.

Style Knowledge Extraction from Memory Prior to generating stylized sentences, we extract the style knowledge according to the concept words of the image. Similar to the memory update operation, we attend to the embedding vectors in $\boldsymbol{M}_s$ and take the weighted-sum of these vectors as the extracted knowledge:

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\boldsymbol{M}_s')^\top \boldsymbol{e}_c, \\
\boldsymbol{\beta} &= \mathrm{softmax}(\hat{\boldsymbol{\beta}}), \\
\boldsymbol{m} &= \boldsymbol{M}_s \boldsymbol{\beta},
\end{aligned} \tag{2}$$

where the vector $\boldsymbol{\beta}$ denotes the weight for each embedding vector when extracting style knowledge. The vector $\boldsymbol{m}$ denotes the extracted style knowledge, which is used to update the hidden state of the captioner.

Memory Based Stylized Captioner

The captioner $\mathcal{C}$ takes a scene graph G and the style knowledge $\boldsymbol{m}$ extracted from $\mathcal{M}$ as input and generate stylized sentence $\hat{y}^s$. The scene graph G is first mapped into a set of embeddings and the extracted style knowledge $\boldsymbol{m}$ initializes the cell state of a two-layer LSTM network.

Encoding Scene Graph We denote the embeddings of object o_i, relationship r_{ij} and attribute a_i^k in a scene graph as $\boldsymbol{e}_{o_i}$, $\boldsymbol{e}_{r_{ij}}$ and $\boldsymbol{e}_{a_i^k}$, respectively. These embeddings are equal to the word embeddings of the nodes' class labels. We further encode the node embeddings to gather context-aware information. Concretely, the context-aware embedding of a relationship r_{ij} is calculated by

$$\boldsymbol{u}_{r_{ij}} = \boldsymbol{W}_{tr}[\boldsymbol{e}_{o_i}; \boldsymbol{e}_{r_{ij}}; \boldsymbol{e}_{o_j}], \tag{3}$$

where $\boldsymbol{e}_{o_i}, \boldsymbol{e}_{r_{ij}}, \boldsymbol{e}_{a_i^k}$ denote node embeddings in the triplet $\langle o_i, r_{ij}, o_j\rangle$, $[;]$ denotes vector concatenation, and $\boldsymbol{W}_{tr} \in \mathbb{R}^{d\times 3d}$ represents a learnable parameter. The context-aware embedding of an object o_i is given by

$$\boldsymbol{u}_{o_i} = \frac{1}{N_i + 1}\left(\sum_{k=1}^{N_i} \boldsymbol{W}_{at}[\boldsymbol{e}_{o_i}; \boldsymbol{e}_{a_i^k}] + \boldsymbol{e}_{o_i}\right), \tag{4}$$

where $\boldsymbol{e}_{a_i^k}$ represents the node embeeding of the k-th attribute of the object o_i, N_i indicates the attribute number of o_i, and $\boldsymbol{W}_{at} \in \mathbb{R}^{d\times 2d}$ is a learnable parameter. The embedding of the whole scene graph is calculated by averaging all the context-aware embeddings, i.e. $\boldsymbol{e}_c = \sum_{p=1}^{K} \boldsymbol{u}_p$, where K denotes the total number of context-aware embeddings.

Generating Stylized Caption The context-aware embeddings $\{\boldsymbol{u}_p|_{p=1}^{K}\}$ of a scene graph are used as the input of top-down attention LSTM to generate a stylized image caption. Specifically, the context-aware embeddings are first attended by the attention LSTM, and the attended embedding is used as the input of the language LSTM. At time step t, the attention weight $\boldsymbol{\gamma}_{t,i}$ of the p-th context-aware embedding $\boldsymbol{u}_p$ is calculated by

$$\begin{aligned}
\bar{\boldsymbol{x}}_t^1 &= [\boldsymbol{h}_{t-1}^2; \boldsymbol{e}_c; \boldsymbol{e}_{w_{t-1}}], \\
\boldsymbol{h}_t^1 &= \mathrm{LSTM}^1(\boldsymbol{h}_{t-1}^1, \bar{\boldsymbol{x}}_t^1), \\
\boldsymbol{\gamma}_{t,p} &= \tanh(\boldsymbol{W}_{va}\boldsymbol{u}_p + \boldsymbol{W}_{ha}\boldsymbol{h}_t^1),
\end{aligned} \tag{5}$$

where $\boldsymbol{W}_{ha}$ and $\boldsymbol{W}_{va}$ are learnable parameters, $\boldsymbol{h}_{t-1}^2$ denotes the previous hidden state of the language LSTM, $\boldsymbol{h}_t^1$ denotes the current hidden state of the attention LSTM, and $\boldsymbol{E}_{w_{t-1}}$ denotes the embedding of the previous word. After calculating the attention weights, the current word w_t is predicted according to the weighted sum of context-aware embeddings:

$$\begin{aligned} \boldsymbol{u} &= \sum_{p=1}^{K} \gamma_{t,p} \boldsymbol{u}_p, \\ \boldsymbol{h}_t^2 &= \mathrm{LSTM}^2(\boldsymbol{h}_{t-1}^2, [\boldsymbol{h}_t^1; \boldsymbol{u}]), \\ \boldsymbol{p}_t &= \mathrm{softmax}(\boldsymbol{W}_o \boldsymbol{h}_t), \\ \hat{w}_t &= \arg\max_{w} p_{t,w}, \end{aligned} \tag{6}$$

where $\boldsymbol{W}_o$ is learnable parameter and $p_{t,w}$ denotes the probability of word w at time step t.

Training Strategy The whole training process of MemCap involves pre-training stage and fine-tuning stage. In pre-training stage, factual data D_f is used to train the captioner. Given a image x and factual sentence y_f, the image scene graph G^x is fed into the captioner. Since style knowledge is not involved in factual sentence, the vector $\boldsymbol{m}$ in Equation 2 is set to all-zero vector. The captioner $\mathcal{C}$ is optimized with cross-entropy loss function:

$$\mathcal{L}_{ce} = -\frac{1}{L} \sum_{i=1}^{L} \log(p(\hat{w}_i = w_i)) \tag{7}$$

The pre-training stage is intended to provide a warm-initialization for the second stage.

In fine-tuning stage, the captioner $\mathcal{C}$ and style memory module $\mathcal{M}$ are trained in an end-to-end manner using unpaired stylized corpus. A stylized training sentence y^s is split into content-related part W_c and style-related part W_s. The scene graph G^y of y^s and the extracted style knowledge $\boldsymbol{m}$ are used as the input of captioner, i.e. $\hat{y}^s = \mathcal{C}(G^y, \boldsymbol{m})$, where $G^y = \mathcal{F}(W_c)$ is the scene graph derived from the content-related part of y^s, and $\hat{y}^s$ is the predicted sentence of captioner. In the first few epochs, the cross-entropy loss in Eq. 7 is used to optimize $\mathcal{C}$ and $\mathcal{M}$. In the rest of the fine-tuning process, we apply REINFORCE (Williams 1992) algorithm with a reward designed for stylized captioning. Denoting the parameters of $\mathcal{C}$ and $\mathcal{M}$ as θ, the gradient of θ is approximated by

$$\nabla_\theta J(\theta) \approx -(r(\hat{y}^s)) \nabla_\theta log_\theta(\hat{y}^s), \tag{8}$$

where $\hat{y}^s$ denotes the sentence acquired by sampling from the probability $\boldsymbol{p}_t$ at each time step . The function $r(\hat{y}^s)$ denotes the reward for sentence $\hat{y}^s$, which contains three components: the CIDEr reward, the style classifier reward and the perplexity reward. Inspired by self-critical training (Rennie et al. 2017) which introduces a baseline for the reward, we define our reward function $r(\hat{y}^s)$ as

$$\begin{aligned} r_{\mathrm{CIDEr}} &= \mathrm{CIDEr}(\hat{y}^s) - \mathrm{CIDEr}(y^*), \\ r_{\mathrm{cls}} &= \mathrm{cls}(\hat{y}^s) - \mathrm{cls}(y^*), \\ r_{\mathrm{ppl}} &= \mathrm{sgn}(-(\mathrm{ppl}(\hat{y}^s) - \mathrm{ppl}(y^*))), \\ r(\hat{y}^s) &= \lambda_1 r_{\mathrm{CIDEr}} + \lambda_2 r_{\mathrm{cls}} + \lambda_3 r_{\mathrm{ppl}} \end{aligned} \tag{9}$$

Algorithm 1 Training Procedure of MemCap

Input: factual dataset $D_f = \{(x_i, y_i^f)\}$, stylized sentence $D_s = \{y_i^s\}$
Output: trained memory module $\mathcal{M}$ and captioner $\mathcal{C}$

procedure PRE-TRAIN($D_f, \mathcal{C}$)
  for (x_i, y_i^s) **in** D_f **do**
    $G^x \leftarrow \mathcal{E}(x_i)$
    $\hat{y}^f = \mathcal{C}(G^x, \mathbf{0})$
    optimize C with Eq.7
  end for
end procedure
procedure FINE-TUNE($D_s, \mathcal{C}, \mathcal{M}$)
  procedure RECONSTRUCT($y^s, \mathcal{C}$)
    split y_i^s into W_s, W_c
    $G^y \leftarrow \mathcal{E}(W_c)$
    update $\boldsymbol{M_s}$ and $\boldsymbol{M_s'}$ with Eq.1
    extract $\boldsymbol{m}$ with Eq.2
    $\hat{y}_i^s \leftarrow \mathcal{C}(G^y, \boldsymbol{m})$
    return $\hat{y}_i^s$
  end procedure
  for y_i^s **in** D_s **do** ▷ training with cross-entropy loss
    $\hat{y}_i^s \leftarrow$ RECONSTRUCT($y_i^s, \mathcal{C}$)
    optimize $\mathcal{C}$ with Eq.7
  end for
  for y_i^s **in** D_s **do** ▷ training with self-critical
    $\hat{y}_i^s \leftarrow$ RECONSTRUCT($y_i^s, \mathcal{C}$)
    optimize $\mathcal{C}$ with Eq.8 and Eq. 9
  end for
end procedure
PRE-TRAIN($D_f, \mathcal{C}$)
FINE-TUNE($D_s, \mathcal{C}, \mathcal{M}$)

where sgn denotes the sign function. The sentence y^* denotes the sentence acquired by taking the word having the maximum probability at each time step, which serves as the baseline for $\hat{y}^s$. The CIDEr reward is calculated according to the ground-truth sentence y^s, which encourages the captioner to preserve the content in the input scene graph. The style classifier reward $\mathrm{cls}(y) \in \{0, 1\}$ is the output of a pre-trained style classifier, indicating whether a sentence expresses the desired linguistic style. The perplexity reward ppl is calculated by a language model pre-trained using sentences with style s. We encourage our model to generate sentences with lower perplexity, since a lower perplexity indicates that the sentence is more fluent. The coefficients λ_1, λ_2 and λ_3 denote the weights of the three components, which are tunable hyper-parameters.

Experiment

Dataset

The factual descriptions and corresponding images are from MSCOCO (Lin et al. 2014) dataset. The stylized descriptions are from SentiCap dataset (Mathews, Xie, and He 2016) that includes positive and negative styles, and FlickrStyle10K dataset (Gan et al. 2017) that includes humorous and romantic styles.

The SentiCap dataset contains 2360 images from MSCOCO dataset, as well as 5013 positive sentences and 4500 negative sentences. For the positive sentences, we use 2994 sentences for training and 2019 sentences for testing, and for the negative sentences, we use 2991 sentences for training and 1509 sentences for testing.

The original FlickrStyle10K dataset is composed of 10,000 images and each image has one romantic description and one humorous description. However, only the official training split that contains 7,000 images is publicly available. Following (Guo et al. 2019), we randomly sample 6,000 images as our training split and the rest images are used for testing.

In all the experiments, the images and sentences in MSCOCO dataset are used to pre-train the captioner $\mathcal{C}$. The stylized sentences in SentiCap dataset and FlickrStyle10K dataset are used for fine-tuning.

Evaluation Metrics

We evaluate our method in two aspects: the ability of preserving the content of the image (relevancy), and the performance of incorporating linguistic styles in the sentence (stylishness), following the practice of (Guo et al. 2019).

To measure the sentence relevancy, the metrics of Bleu-n (Papineni et al. 2002) (including Bleu-1 and Bleu-3), METEOR (Banerjee and Lavie 2005) and CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) are employed.

To evaluate the sentence stylishness, the style classification accuracy (cls) and the average perplexity (ppl) are adopted. The style classification accuracy is measured by the proportion of sentences that correctly reflects the desired style. A logistic regression classifier is trained with both the styled sentences in SentiCap and StyleNet datasets and the factual sentences from MSCOCO dataset. The trained classifier achieves an accuracy of 96%. The average perplexity of all the generated sentences is calculated by a pre-trained language model. Specifically, for each of the four styles, a tri-gram based statistical language model is trained using the SRILM toolkit (Stolcke 2002) and the generated sentences are evaluated by the corresponding language model, respectively. A lower perplexity score means that the generated sentences are more fluent and better reflect the desired linguistic style.

Implementation Details

To generate the scene graph of an image, we employ Faster R-CNN with VGG-16 (Simonyan and Zisserman 2014) backbone. We initialize VGG-16 with weights pre-trained on ImageNet. In the memory module, the size of memory matrices $\boldsymbol{M}_s$ and $\boldsymbol{M}'_s$ are both set to 300×100. In the captioner, the dimension of word embedding vector $\boldsymbol{E}_w$ is set to 300 and the dimensions of cell state of two LSTM layers are set to 512. The values of parameters $\lambda_1, \lambda_2, \lambda_3$ in Equation 9 are set to 1.0, 1.0 and 0.5, respectively. In both pre-training stage and fine-tuning stage, the Adam optimizer (Kingma and Ba 2014) is applied. During pre-training, the learning rate is fixed at 5×10^{-4}. During fine-tuning, the initial learning rate is set to 5×10^{-4} and decays at a rate of 0.8 for every 10 epochs.

Table 1: Results of single-style image captioning. B-n, M and C are the abbreviations for Bleu-n, METEOR and CIDEr, respectively. For metric ppl, the lower value is better. For other metrics, the higher value is better. The styles "pos", "neg", "roman" and "humor" are the abbreviations for positive, negative, romantic and humorous.

method	style	B-1	B-3	M	C	ppl (↓)	cls
SF-LSTM (paired)	pos	50.5	19.1	16.6	60.0	-	-
	neg	50.3	20.1	16.2	59.7	-	-
	roman	27.8	8.2	11.2	37.5	-	-
	humor	27.4	8.5	11.0	39.5	-	-
StyleNet	pos	45.3	12.1	12.1	36.3	24.8	45.2
	neg	43.7	10.6	10.9	36.6	25.0	56.6
	roman	13.3	1.5	4.5	7.2	52.9	37.8
	humor	13.4	0.9	4.3	11.3	48.1	41.9
MemCap	pos	50.8	17.1	16.6	54.4	13.0	99.8
	neg	48.7	19.6	15.8	60.6	14.6	93.1
	roman	21.2	4.8	8.4	22.4	14.4	98.7
	humor	19.9	4.3	7.4	19.4	16.4	98.9

Table 2: Results of multi-style image captioning.

method	style	B-1	B-3	M	C	ppl (↓)	cls
MSCap	pos	46.9	16.2	16.8	55.3	19.6	92.5
	neg	45.5	15.4	16.2	51.6	19.2	93.4
	roman	17.0	2.0	5.4	10.1	20.4	88.7
	humor	16.3	1.9	5.3	15.2	22.7	91.3
MemCap	pos	51.1	17.0	16.6	52.8	18.1	96.1
	neg	49.2	18.1	15.7	59.4	18.9	98.9
	roman	19.7	4.0	7.7	19.7	19.7	91.7
	humor	19.8	4.0	7.2	18.5	17.0	97.1

Results

We compare our MemCap with several state-of-the-art methods for stylized image captioning, including SF-LSTM (Chen et al. 2018), StyleNet (Gan et al. 2017) and MSCap (Guo et al. 2019). SF-LSTM uses paired images and sentences for training, while MSCap and StyleNet can utilize unpaired stylized corpus. Both SF-LSTM and StyleNet are single-style methods, i.e. a model is trained for each style. MSCap is trained under multi-style setting, where a single model is trained to generate sentences in multiple styles. Our MemCap utilizes unpaired stylized corpus, and the evaluation is performed in both single-style and multi-style manners for fair comparison.

Table 1 shows the results of single-style captioning. We have observations as follows:

- Our method substantially outperforms StyleNet with respect to the sentence stylishness (measured by ppl and cls), validating the superiority of the proposed style memory on incorporating linguistic styles into sentences;
- Our method also achieves better results than StyleNet in terms of the sentence relevancy (measured by Bleu-n, CIDEr and METEOR), which verifies that the stylized sentences generated by our MemCap are able to capture

the content of images;

- Despite trained with unpaired stylized corpus, our method still achieves comparable performance to SF-LSTM that uses paired data. Therefore, our method can be readily applied to more application scenarios without the heavy reliance on the paired training data.

The results of multi-style captioning are shown in Table 2. As can be seen from the results, MemCap achieves lower sentence perplexity and higher style accuracy than MSCap for all the styles, validating the superiority of MemCap on multi-style image captioning. Moreover, MemCap outperforms MSCap for most metrics of sentence relevancy, which indicates that the generated stylized sentences by MemCap can still describe the factual content image accurately.

To evaluate our method qualitatively, we show some examples of generated stylized sentences in Figure 3. As illustrated in Figure 3, most generated sentences describe the image content correctly and express the desired linguistic style appropriately. For instance, the words "nice" and "bad" in the first column, as well as the phrases "looking for supremacy" and "to win the game" in the third column, reflect the desired styles evidently.

Ablation Studies

We conduct ablation studies to verify the contribution of each component in single-style setting. The following variants of our full method are evaluated:

- **w/o** $\mathcal{P}$: To verify the effectiveness of the sentence decomposing algorithm, the word-level style labels l_i are replaced with random labels.
- **w/o** $\mathcal{M}$: To evaluate the contribution of the memory mechanism to incorporating linguistic style into sentence, the memory mechanism is removed. The vector $\boldsymbol{m}$ in Equation 2 is replaced by an all-zero vector.
- **w/o sc**: To evaluate the contribution of self-critical training, our MemCap is optimized with only cross-entropy loss in fine-tuning stage.
- **w/o CIDEr**, **w/o ppl**, **w/o cls**: To validate the effect of each reward component in self-critical training , the CIDEr score, perplexity score and style accuracy are removed from the reward in Eq. 9, respectively.

The results of ablation studies are reported in Table 3. From the results, it is interesting to observe that: (1) by removing the sentence decomposer $\mathcal{P}$, the performance on both sentence relevancy and stylishness drops significantly. This indicates that separating the content-related part and the style-related part is necessary to train MemCap. (2) By removing the memory module $\mathcal{M}$, MemCap performs worse on stylishness, validating the importance of the memory module on memorizing and incorporating styles into sentence. (3) When self-critical training is removed, MemCap works worse on both sentence relevancy and stylishness, indicating that the self-critical training is able to improve the captioning performance. When CIDEr is removed from the reward function, the model performs worse in terms of Bleu-3 and CIDEr, verifying that the CIDEr reward contributes to the preserving of visual content. By removing the perplexity reward, the generated sentences have higher perplexity. This indicates that the perplexity reward is helpful for generating more fluent sentences. When the style classifier reward is removed, the style accuracy drops significantly, which proves the contribution of style classifier reward to ensuring the stylishness of the generated sentences.

Table 3: Results of ablation studies on single-style image captioning.

method	style	B-3	C	ppl (↓)	cls
w/o $\mathcal{P}$	pos	15.2	47.0	26.5	63.4
	roman	4.2	19.6	18.3	46.6
w/o $\mathcal{M}$	pos	17.7	54.7	18.6	67.1
	roman	4.3	19.1	23.1	71.2
w/o sc	pos	15.4	46.6	25.6	68.3
	roman	4.3	20.2	22.6	72.4
w/o CIDEr	pos	15.8	46.8	15.2	99.8
	roman	2.9	7.8	22.3	91.3
w/o ppl	pos	16.3	52.0	24.6	99.4
	roman	3.8	17.2	27.0	95.4
w/o cls	pos	18.1	56.4	16.3	65.5
	roman	4.1	17.7	27.2	24.3
MemCap	pos	17.1	54.4	13.0	99.8
	roman	4.8	22.4	14.4	98.7

Extension to Stylized Chinese Video Captioning

We also apply our method to stylized Chinese video captioning on the Youku-VC dataset that contains 1430 short videos from Youku[1], together with roughly 9000 factual Chinese descriptions. The training set, validation set and test set contain 1000, 215 and 215 videos, respectively. The videos together with their corresponding factual descriptions are used as D_f. The stylized corpus D_s is collected by translating and post-editing the sentences in the training sets of SentiCap dataset and FlickrStyle10K dataset. We segment the words in Chinese sentences with the jieba [2] toolkit. The words appearing less than 3 times are pruned, and the size of the vocabulary is 5374.

We show some examples of generated stylized sentences in Figure 4. As can be seen from the results, we observe that most of the sentences generated by MemCap describe the content of the videos correctly and express the desired linguistic style.

Conclusion

We have proposed a MemCap method for stylized image captioning. Our MemCap memorizes the knowledge of linguistic style with a memory module and distills the content-relevant style knowledge with attention mechanism for generating captions. Thus, it generates sentences that describe the content of the image accurately and reflect the desired

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

[2]https://github.com/fxsjy/jieba

Positive: a nice person wearing a dress under a blue umbrella **Negative:** a bad girl in a dress under a umbrella	**Positive:** two beautiful people in the grass eating **Negative:** two stupid people in a field	**Positive:** a nice car parked in front of a building with a clock on it **Negative:** a damaged building with a clock above it	**Humorous:** man at a tennis court looking for supremacy **Romantic:** two players on a tennis court to win the game	**Humorous:** a person is riding a bicycle into the air to catch a fish **Romantic:** a person riding a bike through the air to win the competition

Figure 3: Examples of generated stylized captions. Each column contains an image and corresponding stylized sentences. The styles of the sentences are marked in bold and the words or phrases reflecting the linguistic style are underlined.

Positive: 一个人骑着摩托车在美丽的街道上 (a man is riding a motorcycle on a beautiful road)
Negative: 一个人骑在危险的道路 (a man riding on dangerous road)

Romantic: 一个人骑自行车自去见他的情人（a man is riding a bike to see his lover）
Humorous: 一个人骑自行车在人行道上，害怕迟到（a man is riding a bike on the pavement, afraid of being late）

Figure 4: Examples of generated stylized Chinese video captions. The corresponding English translations are affiliated in the brackets.

linguistic style appropriately. Since MemCap is capable of performing both single-style and multi-style captioning and is trained with unpaired stylized corpus, it can be readily and easily applied to many realistic scenarios. Extensive experiments on two stylized datasets demonstrate the superiority and effectiveness of our method.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant No.61673062 and Alibaba Group through Alibaba Innovative Research Program.

References

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, 382–398. Springer.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6077–6086.

Banerjee, S., and Lavie, A. 2005. 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*, 65–72.

Chen, T.; Zhang, Z.; You, Q.; Fang, C.; Wang, Z.; Jin, H.; and Luo, J. 2018. "factual"or"emotional": Stylized image captioning with adaptive learning and attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 519–535.

Chen, C.-K.; Pan, Z.; Liu, M.-Y.; and Sun, M. 2019. Unsupervised stylish image description generation via domain layer norm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8151–8158.

Gan, C.; Gan, Z.; He, X.; Gao, J.; and Deng, L. 2017. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3137–3146.

Guo, L.; Liu, J.; Yao, P.; Li, J.; and Lu, H. 2019. Mscap: Multi-style image captioning with unpaired stylized text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4204–4213.

Jhamtani, H.; Gangal, V.; Hovy, E.; and Nyberg, E. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 423–430. Association for Computational Linguistics.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Mathews, A. P.; Xie, L.; and He, X. 2016. Senticap: Generating image descriptions with sentiments. In *Thirtieth AAAI conference on artificial intelligence*.

Mathews, A.; Xie, L.; and He, X. 2018. Semstyle: Learning to generate stylised image captions using unaligned text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8591–8600.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; and Black, A. W. 2018. Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7008–7024.

Schuster, S.; Krishna, R.; Chang, A.; Fei-Fei, L.; and Manning, C. D. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, 70–80.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, 6830–6841.

Shin, A.; Ushiku, Y.; and Harada, T. 2016. Image captioning with sentiment terms via weakly-supervised sentiment dataset. In *British Machine Vision Conference*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Stolcke, A. 2002. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4566–4575.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Xu, J.; Sun, X.; Zeng, Q.; Ren, X.; Zhang, X.; Wang, H.; and Li, W. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. *arXiv preprint arXiv:1805.05181*.

Zhang, Y.; Xu, J.; Yang, P.; and Sun, X. 2018. Learning sentiment memories for sentiment modification without parallel data. *arXiv preprint arXiv:1808.07311*.