Chinese Image Captioning based on Long-term Recurrent Convolutional Networks

Weitao Wan
2016310560

Qiao Liu [*]
2016310702

Abstract

Image captioning has been a challenging and important image interpretation task in recent years. We implement a model for image captioning in Chinese language. Our model consists of two parts. The first module is a deep convolutional neural networks (CNN) which takes an image as input and extracts its discriminative features represented in a vector of fixed length. The second module is a recurrent neural network (RNN) which processes the word sequences given the corresponding image features and predicts the word outputs in the next time step. With these two modules. The model is capable of generating proper Chinese descriptions for a given image. We conduct experiments on a dataset annotated by students in Pattern Recognition class which has not been publicly available yet. Results and evaluation showed the effectiveness of the model applied to Chinese image captioning.

1. Introduction

In this paper we implement long-term recurrent convolutional networks(LRCNs)[2]. This model and many other successful models for image captioning were originally applied to English image captioning task. And the most commonly used dataset for image captioning is MS COCO[5] which contains images and corresponding English captions. Our goal is to generate a Chinese sentence describing the main content of a given image. A toy example of the input/output of this task is shown in figure 1. Since there is no suitable public dataset for Chinese image captioning. We annotate Chinese captions for images on our own.

The main difficulty of this task is how to generate proper language descriptions for images. Inspired by machine translation method [1], the image captioning task can be modeled as a sequence learning task. It can be considered as a kind of 'translation' where the source is not a kind of language but an image. So the goal is to 'translate' the image into sentences of target language.

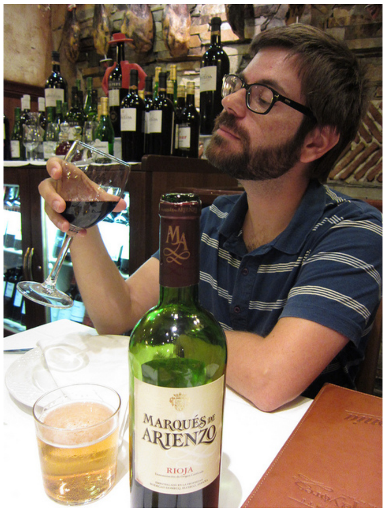

Figure 1. An example image for Chinese image captioning. A proper output can be 'A man is drinking wine in a bar' in Chinese.

Recurrent Neural Networks (RNN) have been explored for their advantages of sequence learning. However, the basic RNN models often suffer from vanishing gradient effect as it strictly integrate state information over time. To learn in a long-range temporal interval, a model called LSTM was first proposed in [3] which contains trainable memory cells to control the input, output and modification of the hidden states. LSTM has proved to be effective for generating proper language descriptions from visual representations derived from CNN models. We extend its application to Chinese image captioning and conduct experiments on our dataset.

2. Method

We implemented the model of LRCN[2] for this task. The whole structure is illustrated in Figure 2, which consists of a deep CNN for image feature extraction and a LSTM language model for sentence generation. The principle of these two modules will be introduced in detail in Sect.2.1 and Sect.2.2 respectively.

[*] Authors contributed equally

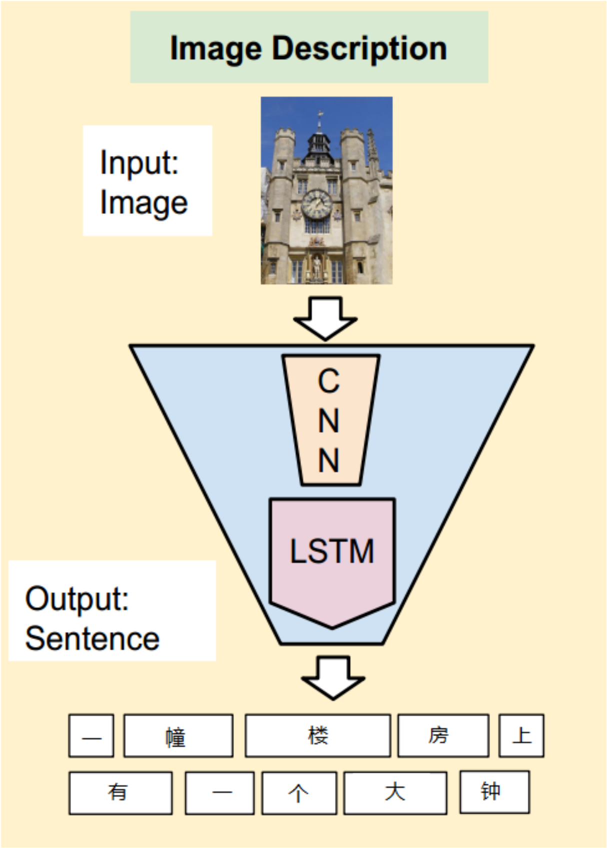

Figure 2. Overview of the whole structure of the Chinese image captioning model.

2.1. Feature Extraction

Deep Convolutional Neural Networks (CNN) have been widely used for extracting discriminative representations of images. CNN typically consists of convolutional layers, down-sampling layers, activation layers and fully connected layers. More sophisticated model may utilize other specially designed layers for different purposes. From a mathematical view, the whole CNN model is a high dimensional non-linear function which is typically defined on $\mathbb{R}^{3\times H\times W}$ for RGB-channel images of height H and width W. And its output is in space $\mathbb{R}^{D}$ where D is the dimension of the final output vectors. Deeper layers intuitively extract higher level features of images.

We use VGG-19[6] for feature extraction. It is a widely used deep CNN model which is pre-trained on the ImageNet dataset. The model structure of VGG-19 is illustrated in Figure 3. It consists of 16 convolutional layers with kernel size 3, 5 down-sampling layers with kernel size 2 and 3 fully connected layers with output dimension 4096, 4096 and 1000 respectively.

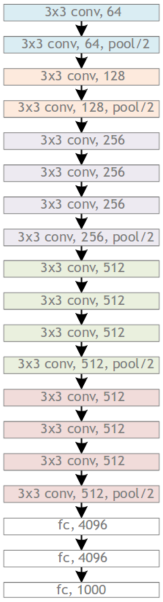

Figure 3. Structure of VGG-19 Convolutional Neural Networks.

2.2. Language Model

In fact, we do not need any explicit language model such as a pre-defined syntax structure in Natural Language Processing (NLP). We train the LSTM model to learn the language structure by supervised learning. Therefore the trained LSTM model is the language model in our whole structure.

The structure of a LSTM unit is illustrated in Figure 5. LSTM processes input sequence $\{x_1, x_2, ..., x_T\}$ of length T, where x is the input word and the subscript is the time step. In each time step t, the LSTM unit takes the input x_t , the hidden state h_{t-1} and the memory cell c_{t-1}. Then it computes the activation values of the gates using its trainable weights and updates the hidden state and the memory cell. The formulations of each update at time step t are given below:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t) + W_{hi}h_{t-1} + b_i && (1)\\
f_t &= \sigma(W_{fi}x_t) + W_{hf}h_{t-1} + b_f && (2)\\
o_t &= \sigma(W_{xo}x_t) + W_{ho}h_{t-1} + b_o && (3)\\
g_t &= \sigma(W_{xc}x_t) + W_{hc}h_{t-1} + b_c && (4)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t && (5)\\
h_t &= o_t \odot \phi(c_t) && (6)
\end{aligned}
$$

The i_t, f_t, o_t and g_t are input gate, forget gate, output gate and input modulation gate respectively. They are determined by the input x_t and the previous hidden state h_{t-1}. The activation function σ is the sigmoid function. The memory cell c_t is determined by the previous memory cell c_{t-1} and the input modulation value g_t, controlled by the

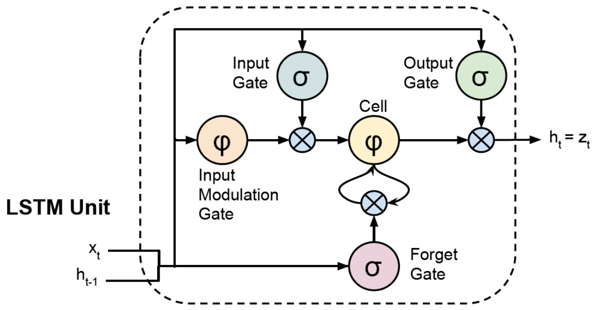

Figure 4. Structure of a LSTM unit, where x is the input and h is the hidden state which is also the output z in test phase. The subscript t is the time step in the sequence.

forget gate and input gate. The hidden state is determined by the memory cell, controlled by the output gate.The operator ⊙ stands for element-wise product. Compared with traditional RNN, these additional gates enable the LSTM to learn extremely complex and long-term temporal dynamics. As is shown in our experiment, the LSTM language model is effective to predict the next word based on the current word and the corresponding image representation.

3. Experiment

Our experiment is conducted upon the dataset built up by students in the 2017 Pattern Recognition class directed by Prof. Changshui Zhang and the TAs. It consists of 8000 images for training, 1000 images for validation and 1000 images for testing. Each image in the train/validation set is annotated with an average of 5 Chinese sentences describing the main content of the image.

We first extract the vocabulary list of the annotated sentences. We experimented two different settings. One is the word-segmentation vocabulary and the other is the single-character vocabulary. For word segmentation, we use the open source model THULAC[4]. After filtering the words which show up for only once, we get a vocabulary of 5735 words. For single-character vocabulary, we simply treat each Chinese character as a unit, ending up with a vocabulary of 2215 characters. In the following of this paper, we will refer to those 'characters' as 'words' since they are the same for word embedding.

We use the one-hot word embedding. Each word in the vocabulary is given an integer index from 1 to N, where N is the vocabulary size. Then each word embedding is a vector $w \in \mathbb{R}^N$ and all the elements are 0 except for $w_k = 1$ where k is the word's index. We use the two-layer, factored architecture of LSTM model as shown in Figure 5. The LSTM has 1000 hidden units.

As is described in Sect. 2.1, we use the VGG-19 model to extract the image features. Specifically, we use the output of the second fully connected layer as our image representations, which is a vector of length 4096. We perform PCA to transform the vector into the length of 1000 to fit the output dimension of the first LSTM layer.

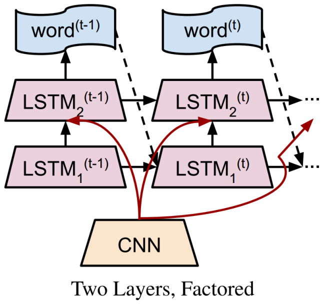

Figure 5. Two LSTMs stacked. The first layer processes the one-hot embeddings of words into better word embeddings. The second layer takes the word embeddings and image representations to predict the next word.

Table 1. Evaluation on the Chinese image captioning test set. 'Single' represents single-char and 'Segment' represents word segmentation.

Method	BLEU1	BLEU2	BLEU3	BLEU4
Single	0.639	0.495	0.375	0.287
Segment	0.596	0.453	0.343	0.261

Table 2. More results

Method	ROUGE	CIDEr
Single	0.484	0.925
Segment	0.462	0.848

We compare the test results of the two models based on the word-segmentation vocabulary and the single-character vocabulary respectively in Table 1. Results can be checked by searching the leader board for our student numbers. The word-segmentation makes the result worse. The reason may be that the vocabulary of the word segmentation is more than twice larger than the one of the single character. And our training set is small. In this case, the single character may perform better.

4. Conclusion

We implemented the model of LRCN for Chinese image captioning and compared two methods, i.e. the single-character method and the word-segmentation method. Experiments showed the effectiveness of the two-layer, stacked version of LSTM model. A main difference between English and Chinese is the word segmentation. Future work is to further explore the word segmentation in Chinese image captioning based on larger dataset. Hopefully a proper word segmentation and enough training data can demonstrate the advantage of word segmentation for Chinese image captioning.

References

[1] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 1

[2] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 1

[3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[4] Z. Li and M. Sun. Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512, 2009. 3

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2