DATA130011.01

Neural Network and Deep Learning

Project III
Novel Image Captioning

Shao Yi

2022-05-30

Abstract

Novel Image Captioning is a challenging task connecting both computer vision and natural language processing. There're a lot of related works on the topic, in this project we will present the results on Decoupled Novel Object Captioner (DNOC), finally we achieve the score of 57.24 on F1, 69.50 on CIDEr-D, 21.68 on METEOR and 14.61 on SPICE.

Repo: https://github.com/Tequila-Sunrise/Decoupled-Novel-Object-Captioner

1 Introduction

1.1 Image Captioning

Image Captioning is the task of describing the content of an image in words. This task lies at the intersection of computer vision and natural language processing. Most image captioning systems use an encoder-decoder framework, in which the Convolutional Neural Network (CNN) is usually used as the image encoder, and the decoder is usually a Recurrent Neural Network (RNN) to sequentially predict the next word given the previous words. The captioning networks need a large number of image-sentence paired data to train a meaningful model. The most popular benchmarks are nocaps and COCO, and models are typically evaluated according to a BLEU or CIDER metric.

1.2 Novel Task and Model Selection

While recent deep neural network models have achieved promising results on the image captioning task, they rely largely on the availability of corpora with paired image and sentence captions to describe objects in context. We would like to address the task of generating descriptions of novel objects which are not present in paired imagesentence datasets.

Certainly, a pre-trained captioning model mentioned above can hardly be applied directly to a brand new domain in which some novel object categories exist, i.e., the objects and their description words are unseen during model training. To correctly caption the novel object, it requires professional human workers to annotate the images by sentences with the novel words, which is time-consuming and labor-expensive and thus limiting its usage in real-world applications.

There are already many works have been published before, including the following: Henzdricks et al. proposed the Deep Compositional Captioner (DCC), a pilot work to address the task of generating descriptions of novel objects which are not present in paired image-sentence datasets. Then Venugopalan et al. discussed a Novel Object Captioner (NOC) to further improve the DCC to an end-to-end system by jointly training the visual classification model, language sequence model, and the captioning model. Anderson et al. leveraged an approximate search algorithm to forcibly guarantee the inclusion of selected words during the evaluation stage of a caption generation model. Yao et al. exploited a mechanism to copy the detection results to the output sentence with a pre-trained language sequence model. What's more, Lu et al. also proposed NBT to generate a sentence template with slot locations, which are then filled in by visual concepts from object detectors.

However, models mentioned above all have to use extra data of the novel object to train their word embedding, in addition, NBT even has to manually defined category mapping list to replace the novel object's word embedding with an existing one. Different from existing methods, DNOC focuses on zero-shot novel object captioning task in which there are no additional sentences or pretrained models to learn such embeddings for novel objects.

2 Decoupled Novel Object Captioner

In this project, we try to solve the zero-shot novel object captioning task where the machine generates descriptions without extra sentences about the novel object. To tackle the challenging problem, we follow the guidance of the Decoupled Novel Object Captioner (DNOC) framework that can fully decouple the language sequence model from the object descriptions. DNOC has two components. The first one is a Sequence Model with the Placeholder (SM-P) to generate a sentence containing placeholders where the placeholder represents an unseen novel object. Therefore, the sequence model can be decoupled from the novel object descriptions. The second one is a key-value object memory built upon the freely available detection model, it contains the visual information and the corresponding word for each object. In short, the SM-P intends to generate a query to retrieve the words from the object memory, and the placeholder will then be filled with the correct word, resulting in a caption with novel object descriptions. The experimental results on the held-out MSCOCO dataset demonstrate the ability of DNOC in describing novel concepts in the zero-shot novel object captioning task.

2.1 Sequence Model with the Placeholder

Sequence Model with the placeholder (SM-P) is applied to fully decouple the sequence model from novel object descriptions. As discussed above, the classical sequence model cannot take an out-of-vocabulary word as input. To solve this problem, a new token is designed, which is denoted as "<PL>". It is used in the decoder similarly to other tokens, such as "<GO>", "<PAD>", "<EOS>", "<UNKNOWN>" in most natural language processing works. And the token "<PL>" is added into the paired vocabulary to learn the embedding.

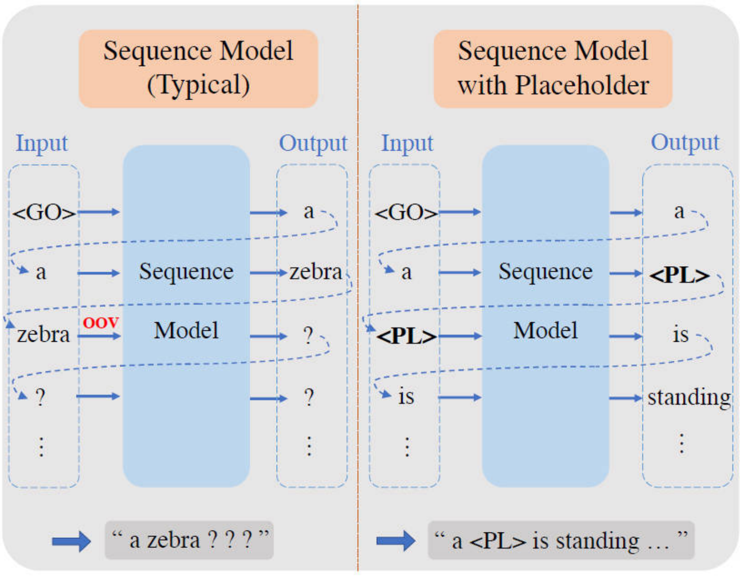


Figure 1: The comparison of the typical sequence model and the proposed SM-P

The new representation could be jointly learned with known words. the new token "<PL>" is carefully designed in both the input and the output of the decoder, which enables us to handle the out-of-vocabulary words. When the decoder outputs "<PL>", the DNOC model utilizes the external knowledge from the object detection model to replace it with novel description. The SM-P is flexible that can be readily incorporated in the sequence to sequence model. Without loss of generality, LSTM is applied as the backbone of SM-P.

For Instance, in Figure **??**, the classical sequence model cannot handle the out-of-vocabulary word "zebra" as input. Instead, the SM-P model outputs the "<PL>" token when it needs to generate a word about the novel object "zebra". This token is further fed to the decoder at the next time step. Thus, the subsequent words can be generated. Finally, the SM-P generates the sentence with the placeholder "A <PL>is standing ...". The "<PL>" token will be replaced by the novel word generated from the key-value object memory.

2.2 Key-Value Object Memory

In order to incorporate the novel words into the generated sentences with the placeholder, DNOC exploits a pre-trained object detection model to build the key-value object memory.

A freely available object detection model is applied on the input images to find novel objects. For the i-th detected object obj j_i, the CNN feature representations $\mathrm{f}_i \in \mathbb{R}^{1\times N_f}$ and the predicted semantic class label $\mathrm{l}_i \in \mathbb{R}^{1\times N_D}$ form a key-value pair, with the feature as key and the semantic label as the value. N_f is the feature dimension of CNN representation, N_D is the number of total detection classes. The key-value pairs associate the semantic class labels (descriptions of the novel objects) with their appearance feature. Futhermore, we extract the CNN feature f_i for obj j_i from the ROI pooling layer of the detection model. Among all the detected results, the top N_{det} key-value pairs are selected according to their confidence scores, which form the key-value object memory $\mathcal{M}_{\mathrm{obj}}$. For each input image, the memory $\mathcal{M}_{\mathrm{obj}}$ is initialized to be empty.

Let $\mathcal{W}_{\mathrm{det}}$ be the vocabulary of the detection model, which consists of N_D detection class labels. Note that each word in $\mathcal{W}_{det}$ is the detection class label in the one-hot format, since we cannot obtain trained word embedding function $\phi_w(\cdot)$ for the new word. To generate the novel word and replace the placeholder in the sentence at time step t, we define the query q_t to be a linear transformation of previous hidden state h_{t-1} when the model meets the special token "<PL>" at time step t:

$$q_t = w_{t-1},$$

where $\mathbf{h}_{t-1} \in \mathbb{R}^{N_h}$ is the previous hidden state at ($t-1$)-th step from the sequence model, and $w \in \mathbb{R}^{N_f\times N_h}$ is the linear transformation to convert the hidden state from semantic feature space to CNN appearance feature space. We have the following operations on the key-value memory $\mathcal{M}_{\mathrm{obj}}$:

$$\begin{aligned} \mathcal{M}_{\mathrm{obj}} &\leftarrow \mathrm{WRITE}\left(\mathcal{M}_{\mathrm{obj}}, (\mathrm{f}_i, \mathrm{l}_i)\right) \\ \mathrm{w}_{\mathrm{obj}} &\leftarrow \mathrm{READ}\left(q, \mathcal{M}_{\mathrm{obj}}\right) \end{aligned}$$

WRITE operation is to write the input key-value pair $(\mathrm{f}_i, \mathrm{l}_i)$ into the existing memory $\mathcal{M}_{\mathrm{obj}}$. The input key-value pair is written to a new slot of the memory.

READ operation takes the query q as input, and conducts content-based addressing on the object memory $\mathcal{M}_{\mathrm{obj}}$. It aims to find related object information according to the similarity metric, qK^T. The output of READ operation is,

$$\mathrm{w}_{\mathrm{obj}} = \left(qK^T\right) V$$

where $K^T \in \mathbb{R}^{N_f\times N_{\mathrm{det}}}, V \in \mathbb{R}^{N_{\mathrm{det}}\times N_D}$ are the vertical concatenations of all keys and values in the memory, respectively. The output $\mathrm{w}_{\mathrm{obj}} \in \mathbb{R}^{N_D}$ is the combination of all semantic labels. In evaluation, the word with the max prediction is used as the query result.

2.3 Framework Overview

With the above two components, DNOC framework is proposed to caption images with novel objects. The framework is based on the encoder-decoder architecture with the SM-P and key-value object memory. For an input image with novel objects, we have the following steps to generate the captioning sentence:

1. We first exploit the SM-P to generate a captioning sentence with some placeholders. Each placeholder represents an unseen word/phrase for a novel object;
2. We then build a key-value object memory $\mathcal{M}_{\text{obj}}$ for each input based on the detection feature-label pairs $\{\text{f}_i, l_i\}$ on the image;
3. Finally, we replace the placeholders of the sentence by corresponding object descriptions. For the placeholder generated at time step t, we take the previous hidden state $\mathbf{h}_{t-1}$ from SM-P as a query to read the object memory $\mathcal{M}_{\text{obj}}$, and replace the placeholder by the query results w_{obj}.

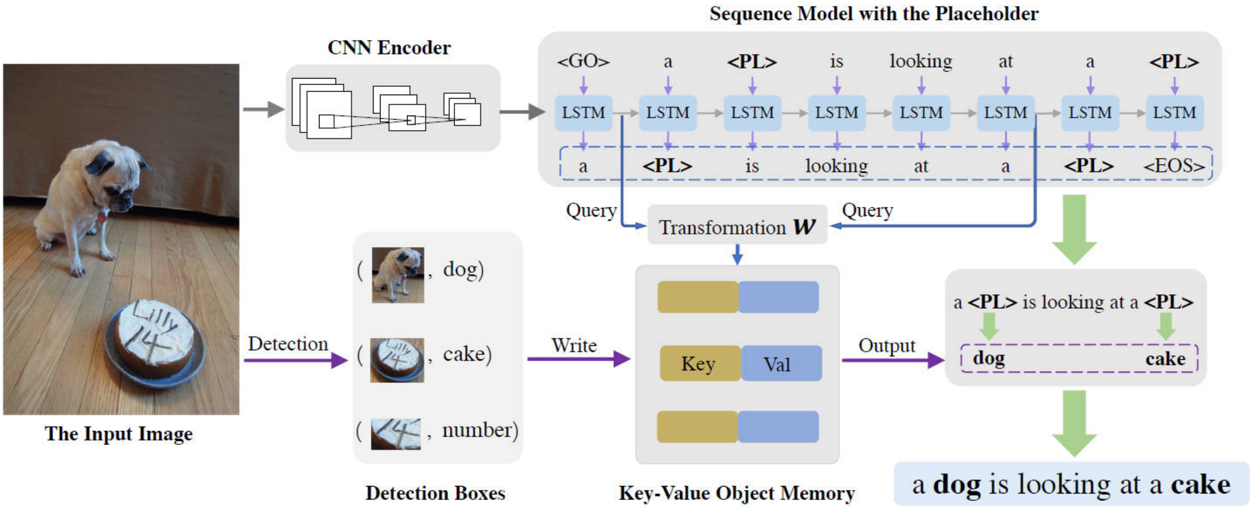


Figure 2: The overview of the DNOC framework

In the example shown in Figure **??**, the “dog” and “cake” are the novel objects which are not present in training. The SM-P first generates a sentence “a <PL>is looking at a <PL>”. Meanwhile, we build the key-value object memory $\mathcal{M}_{\text{obj}}$ based on the detection results, which contains both the visual information and the corresponding word (the detection class label). The hidden state at the step before each placeholder is used as the query to read from the memory. The memory will then return the correct object description, i.e., “dog” and “cake”. Finally, we replace the placeholders by the query results and thus generate the sentence with novel words “a dog is looking at a cake”.

2.4 Training Details

To learn how to exploit the ”out-of-vocabulary” words, we modify the input and target for SM-P in training. We define $\mathcal{W}_{pd}$ as the intersection set of the vocabulary $\mathcal{W}_{\text{paired}}$ and vocabulary $\mathcal{W}_{\text{det}}$,

$$\mathcal{W}_{pd} = \mathcal{W}_{\text{paired}} \cap \mathcal{W}_{\text{det}}$$

Then we modify the input annotation sentence of the sequence model SM-P by replacing each word $w_i \in \mathcal{W}_{pd}$ with the token "<PL>". The new input word $\hat{\mathbf{w}}_t$ at t-th time step is,

$$\hat{\mathbf{w}}_t = \begin{cases} \langle PL \rangle, & \mathbf{w}_t \in \mathcal{W}_{pd} \\ \mathbf{w}_t, & \text{otherwise} \end{cases}$$

The loss function is composed of two parts. The first part (for SM-P) is defined as:

$$\begin{aligned} &\mathcal{L}_{SM-P}\left(\hat{\mathbf{w}}_0, \hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{t-1}, \phi_e(\mathrm{I}); \theta_{SM-P}\right) = \\ &-\sum_t \log\left(\operatorname{softmax}_t\left(F_{SM}\left(\hat{\mathbf{w}}_0, \hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{t-1}, \phi_e(\mathrm{I}); \theta_{SM-P}\right)\right)\right) \end{aligned}$$

where the $\operatorname{softmax}_t$ denotes the softmax operation on the t-th step.

As for the key-value object memory $\mathcal{M}_{\text{obj}}$, we define the optimizing loss by comparing the query result $\mathbf{w}_{obj_t}$ from object memory and the word $\mathbf{w}_t$ from annotation,

$$\mathcal{L}_{\mathcal{M}_{\text{obj}}} = \sum_t a_t CE\left(\mathbf{w}_{obj_t}, \mathbf{w}_t\right)$$

where $CE(\cdot)$ is the cross-entropy loss function, and a_t is the weight at time step t that is calculated by,

$$a_t = \begin{cases} 1, & \mathrm{w}_t \in \mathcal{W}_{pd} \\ 0, & \text{otherwise.} \end{cases}$$

There are two trainable components above. One is the query q, the hidden state from the LSTM model. The other is the linear transformation on detection features in the computation of the memory key. We simultaneously minimize the two loss functions.

The final objective function for the DNOC framework is,

$$\mathcal{L} = \mathcal{L}_{SM-P} + \mathcal{L}_{\mathcal{M}_{\text{obj}}}$$

3 Experiment Results

3.1 The held-out MSCOCO dataset

The MSCOCO dataset is a large scale image captioning dataset. For each image, there are five human-annotated paired sentence descriptions. Following the previous works, we employ the subset of the MSCOCO dataset, but excludes all image-sentence paired captioning annotations which describe at least one of eight MSCOCO objects. The eight objects are chosen by clustering the vectors from the word2vec embeddings over all the 80 objects in MSCOCO segmentation challenge. It results in the final eight novel objects for evaluation, which are "bottle", "bus", "couch", "microwave", "pizza", "racket", "suitcase", and "zebra". These novel objects are held-out in the training split and appear only in the evaluation split.

3.2 Experimental Settings

Evaluation Metrics. Metric for Evaluation of Translation with Explicit Ordering (METEOR) is an effective machine translation metric which relies on the use of stemmers, WordNet synonyms and paraphrase tables to identify matches between candidate sentence and reference sentences. However, as pointed before, the METEOR metric is not well designed for the novel object captioning task. It is possible to achieve high METEOR scores even without mentioning the novel objects. Therefore, to better evaluate the description quality, we also use the F1-score as an evaluation metric. F1-score considers false positives, false negatives, and true positives, indicating whether a generated sentence includes a new object. In addition, metrics like CIDEr-D and SPICE are also used to evaluate the quality of the generated descriptions.

Implementation Details. We apply a 16-layer VGG pre-trained on the ImageNet ILSVRC12 dataset as the visual encoder. The CNN encoder is fixed during model training. The decoder is an LSTM with cell size 1,024 and 15 sequence steps. For each input image, we take the output of the fc7 layer from the pre-trained VGG-16 model with 4,096 dimensions as the image representation. The representations are processed by a fullyconnected layer and then fed to the decoder SM-P as the initial state. For the word embedding, we do not exploit the per-trained word embeddings with additional knowledge data. Instead, we learn the word embedding ϕ_w with 1,024 dimensions for all words including the placeholder token. We implement our DNOC model with TensorFlow. Our DNOC is optimized by ADAM with the learning rate of 1×10^{-3}. The weight decay is set to 5×10^{-5}. We train the DNOC for 20 epochs and choose the model with the best validation performance for testing.

3.3 Performance Results

We compare DNOC with some other state-of-the-art methods on the held-out MSCOCO dataset.

Method	F_{bottle}	F_{bus}	F_{couch}	$F_{\text{microwave}}$	F_{pizza}	F_{racket}	F_{suitcase}	F_{zebra}	F_{average}
DCC	4.63	29.79	45.87	28.09	64.59	52.24	13.16	79.88	39.78
NOC	14.93	68.96	43.82	37.89	66.53	65.87	28.13	88.66	51.85
LSTM-C	29.68	74.42	38.77	27.81	68.17	**70.27**	44.76	**91.4**	55.66
NBT+G	7.1	73.7	34.4	**61.9**	59.9	20.2	42.3	88.5	48.5
DNOC(Paper)	**33.04**	76.87	53.97	46.57	**75.82**	32.98	59.48	84.58	**57.92**
DNOC(Our)	27.85	**77.77**	**54.47**	48.63	75,34	29.66	**60.10**	84.13	57.24

Table 1: Comparison on F1-score

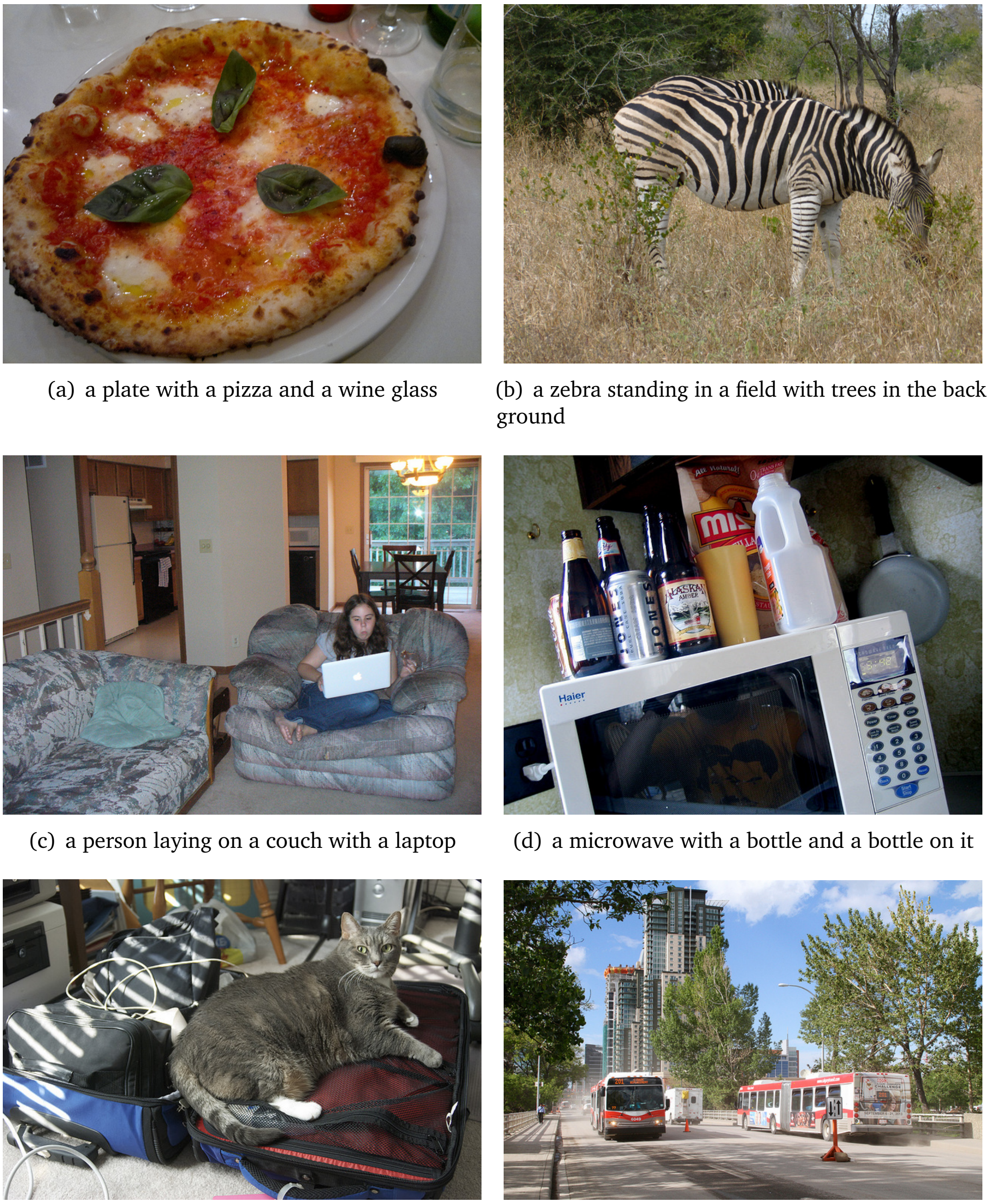
The other evaluation metrics are presented in the following table.

Method	CIDEr-D	METEOR	SPICE
DNOC(Paper)		20.41	
DNOC(Our)	69.50	21.68	14.61

Table 2: Other evaluation metrics

3.4 Qualitative Results

In Figure **??** we will show some qualitative results on the held-out MSCOCO dataset.



(a) a plate with a pizza and a wine glass

(b) a zebra standing in a field with trees in the background

(c) a person laying on a couch with a laptop

(d) a microwave with a bottle and a bottle on it

(e) a cat laying on top of a suitcase

(f) a bus is driving down the street in the city

Figure 3: Qualitative results

4 Conclusion

In this project, we reproduced the work of DNOC on the held-out MSCOCO dataset, the experiment was carried out successfully and the results are quite good.

As for the further work, attention-based image captioning is a promising research area, as some well-known models are already implemented in the literature and worth to be studied.