Mathematical Problems of Computer Science 48, 42–49, 2017.

Image Caption Generation and Object Detection via a Single Model

Aghasi S. Poghosyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: agasy18@gmail.com

Abstract

Automated semantic information extraction from the image is a difficult task. There are works which can extract image caption or object names and their coordinates. This work presents a merged single model of object detection and automated caption generation systems. The final model extracts from image caption and object coordinates with their names without losing accuracy according to initial models.

Keywords: Neural networks, Image caption, Object detection, Deep learning, RNN, LSTM

1. Introduction

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. The content can be partially described via image caption and objects names and their locations.

This is significantly harder than the well-studied image classification [1] or object recognition. These studies can help visually impaired people better understand the content of images on the Web, also it can have a great impact on search engines and in robotics, for example, self driving cars.

Automatically generated image caption should contain the main object names, their properties, relations, and actions. Moreover, the generated caption should be expressed through a natural language like English. There are a number of works approaching this problem. Some of them [2, 3, 4] offer combining the existing image object detection and sentence generation systems. But there is a more efficient solution [5] that offers a joint model. It takes an image and generates the caption, which describes the image adequately. The lastest achievements in statistical machine translation were actively used in image caption generation tasks. The reason for this is mainly the proven achievement of greater results when using a powerful sequential model trained by maximizing the probability of the correct translation for the input sentence. These models [6, 5, 7] are based on Recurrent Neural Networks (RNNs). The model encodes the variable length input into the fixed length vector representation. This representation enables conversion of the input sentence into the target sentence or the input image into the target image caption.

Neural nets have become a leading method for high quality object detection in recent years. Modern object detectors based on Convolutional Neural Network (CNN) [8] networks, such as Faster Region-based Convolutional Neural Network (Faster R-CNN) [9], Region-based Fully Convolutional Network (R-FCN) [10], Multibox [11], Single Shot Detector (SSD) [12] and YOLO: Real-Time Object Detection [13], are now good enough to be deployed in consumer products (e.g., Google Photos, Pinterest Visual Search) and some of them have been shown to be fast enough to run on mobile devices.

There is work [14] which present a multi-model neural network method closely related to the human visual system that automatically learns to describe the content of images. The model consists of two sub-models: an object detection and localization model, which extract the information of objects and their spatial relationship in images respectively; besides, a deep recurrent neural network (RNN)-based on Long Short-Term Memory (LSTM) units with an attention mechanism for sentences generation. Each word of the description is automatically aligned to different objects of the input image when it is generated. It is similar to the attention mechanism of the human visual system.

This work present a merged model of object detection and automated caption generation systems. For object detection we will choose Faster R-CNN [9] based on Inception [15] and for caption generation Show and Tell [5]. These two models are based on Inception image classification model. This will allow as save all quality characteristics of the initial models.

2. Object Detection

The R-CNN paper by Girshick et al. [16] was among the first modern incarnations of convolutional network-based detection. Inspired by recent successes in image classification [17], the R-CNN method took a straightforward approach of cropping externally computed box proposals out of an input image and running a neural net classifier on these crops. This approach can be expensive, however, because many crops are necessary, leading to significant duplicated computation from overlapping crops. Fast R-CNN [9] alleviated this problem by pushing the entire image once through a feature extractor then cropping from an intermediate layer so that crops share the computation load of feature extraction.

In the Faster R-CNN detection happens in two stages. At the first stage, called the *region proposal network* (RPN), images are processed by a feature extractor, and features at some selected intermediate level are used to predict class-agnostic box proposals.

$$L(a, I; \theta) = \alpha \cdot 1[a \text{ is positive}] \cdot \ell_{loc}(\phi(b_a; a) - f_{loc}(I; a, \theta)) + \beta \cdot \ell_{cls}(y_a, f_{cls}(I; a, \theta)), \quad (1)$$

The loss function for this first stage takes the form of Equation 1 using a grid of anchors tiled in space, scale and aspect ratio. At the second stage, these (typically 300) box proposals are used to crop features from the same intermediate feature map which are subsequently fed to the remainder of the feature extractor in order to predict a class and class-specific box refinement for each proposal. The loss function for this second stage box classifier takes the form of Equation 1 using the proposals generated from the RPN as anchors. Notably, one does not crop proposals directly from the image and re-run crops through the feature extractor, which would be a duplicated computation. However, there is a part of computation that must be performed once per region, and, thus, the run time depends on the number of regions proposed by the RPN.

Determining classification and regression targets for each anchor requires matching anchors to groundtruth instances. Common approaches include greedy bipartite matching (e.g.,

based on Jaccard overlap) or many-to-one matching strategies in which bipartiteness is not required, but matchings are discarded if Jaccard overlap between an anchor and groundtruth is too low. Paper [18] refers to these strategies as *Bipartite* or *Argmax*, respectively. The model [18] uses *Argmax* matching throughout with thresholds set as suggested in the original paper [9]. After matching, there is typically a sampling procedure designed to bring the number of positive anchors and negative anchors to some desired ratio.

To encode a groundtruth box with respect to its matching anchor, the model uses the box encoding function $\phi(b_a; a) = [10 \cdot \frac{x_c}{w_a}, 10 \cdot \frac{y_c}{h_a}, 5 \cdot \log w, 5 \cdot \log h]$ (also used by [16, 9]). The scalar multipliers 10 and 5 are typically used in all of these prior works [18, 9, 16].

For our work we will use pretrained Faster-RCNN based on Inception classifier [18]. We will extract high level features before object detector. Extracted features will be used to create an image embedding vector.

3. Caption Generaton

The model encodes the variable length input into the fixed length vector representation. This representation enables conversion of the input sentence into the target sentence or the input image into the target image caption. The last model was being trained to maximize $P(S|I)$ likelihood to generate the target sequence of words $S = \{S_1, S_2, ...\}$ for an input image I, where each word S_t comes from a given dictionary, that describes the image adequately. Show and Tell [5] model can generate image descriptions with recurrent neural network. It maximizes the probability of the correct caption for the given image,

$$\log p(S|I; \theta) = \sum_{t=0}^{N} \log p(S|I, S_0, ..., S_{t-1}; \theta), \tag{2}$$

where $(S|I)$ is a training example pair. While training, we optimize the sum of the log probabilities for the whole training set using AdaGrad [19].

$p(S|I, S_0, ..., S_{t-1}; \theta)$ probability will correspond to the t step (iteration) of Recurrent Neural Network (RNN) based model. The variable number of words that are conditioned upon, up to $t-1$ is expressed by a fixed length hidden state or memory h_t. After every iteration for the new input, memory will be updated by using a non-linear function f.

$$f_{t+1} = f(h_t, x_t). \tag{3}$$

In this work, we will select *Mixed_7c* layer from Google Inception [15] (we will use Object detector's Inception [18]) and append *average pooling* layer which will have 2048-dimensional output for image description. Also, we will append *fully connected* neural layer with N_e neurons, which will convert 2048-dimensional vector into N_e dimensional vector. N_e is an image-words embedding vectors dimensionality [20]. The output vector x_1 of fully connected layer will be the first feed vector for RNN,

$$x_{-1} = Mixed_{7c} * W_i + b_i, \tag{4}$$

where $W_i \in R^{2048xN_e}$ and $b_i \in R^{N_e}$ are trainable parameters for image embedding. We also have lookup embedding matrix $W_l \in R^{DxN_e}$, where D is the dictionarys words count. Each row of the matrix represents a word embedding in image-word embedding space. Each x_i (where $i \geq 0$) is the corresponding row at index (S_i)(Equation 5).

$$x_i = W_e^{S_i}. \tag{5}$$

For f from Equation 3 we use a Long Short-Term Memory (LSTM) [21], which has shown state-of-the-art performance on sequence generation tasks, such as translation or image caption generation.

Long Short-Term Memory (LSTM) is an RNN cell. It helps in solving RNN training time problems like vanishing and exploding gradients [21], which is a significant problem for RNNs. LSTM is commonly used in machine translation, sequence generation and image description generation tasks. Paper [5] uses recurrent neural network with an LSTM cell to generate image caption. From a construction perspective, LSTM is a memory cell c encoding knowledge at every iteration of what inputs have been seen up to this iteration. Later this knowledge is used for subsequent word generation (10, 11). Behavior of the cell is controlled by three gates: an *input gate*, an *output gate* and a *forget gate*. Each gate is a vector of real number elements ranging from 0 to 1. In particular, the *forget gate* is responsible for controlling whether to forget the cells old value, the *input gate* controls the permission for reading a new input value and finally the *output gate* controls the permission to output the new value from the cell. This is done by multiplying the given gate with the corresponding value (9, 10). The definition of the LSTM is as follows:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}), \tag{6}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}), \tag{7}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}), \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}), \tag{9}$$

$$m_t = o_t \odot c_t, \tag{10}$$

$$p_{t+1} = softmax(W_{pm} * m_t), \tag{11}$$

In (6)-(11) equations i_t, o_t, f_t are *input*, *output* and *forget* gates, correspondingly, c_t is a cell memory in step t and m_t is an output of the LSTM for step (iteration) t. W_{ix}, W_{im}, W_{fx}, W_{fm}, W_{ox}, W_{om}, W_{cx}, W_{cm} are trainable parameters (variables) of the LSTM.$\odot$ represents the product with a gate value. Sigmoid $\sigma(\cdot)$ and $h(\cdot)$ hyperbolic tangent are nonlinearities of the LSTM. Equation 11 will produce a probability distribution p_{t+1} over all words in the dictionary, where W_{pm} is a trainable parameter.

The LSTM model is trained to predict the probability for the next word of an image caption after it has observed all the previous words in the captions and image features. For easier training LSTM is represented in unrolled form, which is a copy of the LSTM memory for the image and each word of the sentence. Also all LSTMs share the same parameters.

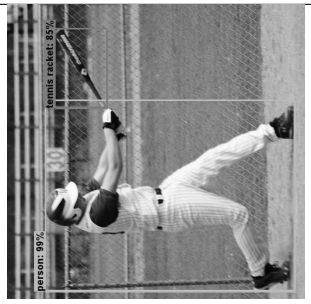
Thus, x_{-1} is the first input for the first LSTM. Initial state of the LSTM is c_{-1} zero-filled memory. For the next LSTMs, inputs correspond to the word embedded vectors. Also, all recurrent connections are converted into feed-forward connections. Loss function will be sum of the negative log likelihood of the correct word at each step:

$$L(I, S) = -\sum_{t=1}^{N} \log p(S_t).$$

For training, we have used AdaGrad instead of multi-batch stochastic gradient descent. We have trained on Microsoft Common Objects in Context (MSCOCO) [22] image dataset and keep the same metrics from the original work in caption generation task. [5].

Inference has been made via using Beam Search which gives us variants for the best scored sentence after many predictions in Table 1.

Table 1: Image caption generation and object detection via a single model.

	1) a group of elephants standing next to each other . 2) a group of elephants standing in a pen . 3) a group of elephants standing next to each other in a zoo.
	1) a group of people standing on top of a sandy beach . 2) a group of people standing on top of a beach . 3) a group of people standing on a beach next to the ocean .
	1) a baseball player holding a bat on a field . 2) a baseball player swinging a bat on a field . 3) a baseball player swinging a bat at a ball
	1) a man sitting in a chair holding a banana . 2) a man sitting in a chair holding a hot dog . 3) a man sitting in a chair holding a banana

4. Conclusion

We have built an image caption generation model on top of object detection model. As our chosen object detection model has same object classifier (feature extractor) as our chosen captions generation model, we have the kept all accuracy metrics from both initial works. Thus, we have created a single model that can generate an image caption and detect objects names and their locations.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[2] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *European conference on computer vision*. Springer, 2010, pp. 15–29.

[3] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Babytalk: Understanding and generating simple image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[4] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

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

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[8] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[10] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[11] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," *arXiv preprint arXiv:1412.1441*, 2014.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[14] Z. Yang, Y.-J. Zhang, Y. Huang et al., "Image captioning with object detection and localization," arXiv preprint arXiv:1706.02430, 2017.

[15] C. Szegedy, V. Vanh" *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[16] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", *in Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[17] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *in Advances in neural information processing systems*, pp. 10971105, 2012.

[18] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama et al., "Speed/accuracy trade-offs for modern convolutional object detectors", arXiv preprint arXiv:1611.10012, 2016.

[19] M. D. Zeiler, "Adadelta: an adaptive learning rate method", arXiv preprint arXiv:1212.5701, 2012.

[20] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781, 2013.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

Submitted 30.08.2017, accepted 05.12.2017.

Պատկերի վերնագրի գեներացումը և օբյեկտների հայտնաբերումը մեկ մոդելի միջոցով

Ա. Պողոսյան

Ամփոփում

Պատկերի մասին իմաստաբանական ինֆորմացիայի ավտոմատացված ստացումը բարդ խնդիր է: Կան աշխատանքներ, որոնք գեներացնում են պատկերի վերնագիրը կամ գտնում օբյեկտների կոորդինատները վերջիններիս անվանումների հետ մեկտեղ: Այս աշխատանքը ներկայացնում է մեկ ամբողջական մոդել, որը կարողանում է գեներացնել պատկերի վերնագիրը և օբյեկտների անվանումները իրենց կոորդինատներով:

Генерация заголовка изображения и обнаружение объекта с помощью единой модели

А. Погосян

Аннотация

Автоматическое извлечение семантической информации из изображения - сложная задача. Существуют работы, которые могут извлекать заголовки изображений или имена объектов и их координаты. Эта работа представляет собой объединенную единую модель обнаружения объектов и автоматического формирования заголовков.