An Improved Image Attention Annotation Algorithm Integrating Spatial Features: Postprint

Authors: Xu Shoukun, Zhou Jia, Li Ning, stone forest

Date: 2018-05-20T00:00:00+00:00

Abstract

To address the issues of insufficient feature selection in the integration of image captioning and attention mechanisms, as well as the inadequate weighting of spatial features during prediction, we propose an attention-based image captioning method that incorporates spatial features. First, image features are obtained through a convolutional neural network, with feature regions aligned to the textual caption sequence. Then, the attention mechanism is employed to weight the caption words, and by combining a spatial feature extraction loss function, we derive spatial feature attention-based image captioning. Finally, validation is performed on both the Flickr30k and COCO datasets, where visualization demonstrates how the model automatically learns salient regions and generates corresponding word output sequences. Experimental results show that the method can effectively extract attention regions and produce captions, achieving better captioning results compared with other models.

Full Text

Improved Algorithm for Image Attention Annotation Combined with Spatial Features

Xu Shoukun1†, **Zhou Jia**1, **Li Ning**1,2, **Shi Lin**1

1. School of Mathematics & Physics, School of Information Science & Engineering, Changzhou University, Changzhou, Jiangsu 213164, China;
2. Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang College), Fuzhou 350108, China

Abstract: To address the problems of insufficient feature selection and inadequate weighting of spatial features during the integration of image annotation and attention mechanisms, this paper proposes an attention-based image annotation method that incorporates spatial features. First, image features are extracted through a convolutional neural network, with feature regions matched

to textual annotation sequences. Then, the attention mechanism weights annotation vocabulary, and a loss function incorporating spatial feature extraction yields image annotations based on spatial feature attention. Finally, validation is performed on both Flickr30k and COCO datasets, with visualizations demonstrating how the model automatically learns salient regions and generates corresponding output vocabulary sequences. Experimental results show that this method can effectively extract attention regions and produce annotations, achieving better annotation performance compared to other models.

Keywords: visual attention; image annotation; spatial feature

0 Introduction

The successful application of sequence-to-sequence encoder-decoder frameworks in machine translation [1] has provided better implementation and utility for the image annotation domain. Kiros et al. [2] proposed a multimodal log-bilinear model using feedforward neural networks to predict the next annotation word. Vinyals et al. [3] used LSTM instead of RNN as the decoder, employing the output of CNN fully connected layers for image annotation. Karpathy et al. [4] derived annotation rankings and joint embedding spaces from R-CNN object detection results and bidirectional RNN outputs. In recent years, attention mechanisms have been introduced into encoder-decoder neural frameworks, yielding improved image annotation performance. Originating from machine translation, attention mechanisms incorporate human neural attention factors into image annotation, enabling better information extraction and annotation. Xu et al. [5] applied attention mechanisms to generate image-aligned words, proposing a model that combines LSTM hidden states with visual attention, which represents a relatively mature attention-based image annotation model to date. Yang et al. [6] extended the current attention encoder-decoder framework by adding verification networks and incorporating vectors that capture global attributes into the decoder mechanism. You [7] and Wu et al. [8] addressed semantic-geometric image visual attention attribute problems using LSTM inputs or outputs, also achieving promising results [9,10].

The proposed method primarily extracts image features through convolutional neural network training while increasing the weight of spatial feature factors within the network. Using an LSTM model with attention mechanism as the encoder-decoder, image annotation is performed through attention weighting combined with spatial features to obtain image annotation results based on spatial feature attention. Finally, visualization demonstrates the attention weights, annotation results, and their analysis.

1 Related Concepts

1.1 Attention Encoder-Decoder

The Attention Model is a brain attention simulation whose core is the Encoder-Decoder process. The Encoder-Decoder model is a classic natural language processing architecture where the Encoder module encodes the input sequence into a code, which is then fed to the Decoder module to produce a specific output sequence. [Figure 1: see original paper] illustrates the general Encoder-Decoder framework.

The input is typically a sequence $X = \{x_1, x_2, ..., x_n\}$, and the output is a sequence $Y = \{y_1, y_2, ..., y_m\}$. In the Encoder module, the input sequence is encoded, represented by $C = F(x_1, x_2, ..., x_n)$. In the Decoder module, C is decoded to compute output y_i using both C and previously generated $y_1, y_2, ..., y_{i-1}$, expressed as $y_i = G(C, y_1, y_2, ..., y_{i-1})$. Notably, when computing output y_i in the Decoder module, the semantic information used is identical. For longer input sequences, limited semantic encoding vector dimensions cause partial effective information loss.

Introducing the Attention Model mechanism into Encoder-Decoder involves two computational processes: calculating semantic encoding under attention probability distribution and computing feature vectors. The specific steps are:

a) **Computing semantic encoding of attention distribution probabilities**: The main idea is to calculate relationship scores between historical nodes and the final input node, then compute their proportion of the total score. The following formulas yield the attention probability of each input relative to the final input:

The attention probability weight α_{ij} represents the attention probability of node x_j for node y_i:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

where $e_{ij} = \tanh(W_h h_{i-1} + U_h x_j + b)$, W_h, U_h, and b are weight matrices and bias, and h_{i-1} is the hidden state corresponding to the final input.

b) **Computing semantic encoding and feature vectors under attention distribution**: The semantic encoding C_i is obtained through the cumulative sum of attention probability weights multiplied by hidden states of historical input nodes. The final semantic encoding combines the attention probability distribution semantic encoding with the overall article vector as input to the traditional LSTM module, where the final node' s hidden state value h_T serves as the ultimate feature vector. This feature vector contains weight information from historical input nodes, emphasiz-

ing keyword semantic information [11]:

$$C_i = \sum_{j=1}^{T} \alpha_{ij} x_j$$

[Figure 2: see original paper] shows the framework of the Encoder-Decoder model with Attention Model. Each output element y_i has a corresponding semantic encoding C_i from the input sequence probability distribution. For output y_i, the calculation is:

$$y_i = G(\{x_1, x_2, ..., x_n\}, \{y_1, y_2, ..., y_{i-1}\}, C_i)$$

where $C_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$ represents the semantic encoding at time i, α_{ij} is the attention probability of input x_j for output y_i, and T_x is the number of elements in the input sequence. This design computes attention probability distributions for the current output y_i, yielding unique semantic encoding information that fuses input attention distributions to optimize current output.

1.2 Spatial Features

Convolutional neural networks typically extract image features through multiple serial convolutional layers and pooling layers arranged alternately to learn image data features layer by layer. Convolution operations use kernels smaller than the image size to scan the entire image, computing the weighted sum of the kernel and local image positions. Each convolution corresponds to a feature map subsequently fed to pooling layers for spatial subsampling, giving CNNs certain distortion resistance. The network' s topmost layer flattens all obtained feature maps into one-dimensional vectors combined with multi-class regression classifiers, backpropagating error signals to adjust network parameters.

Spatial features constitute an important part of object spatial judgment ability in static images. A key characteristic of image data is the obvious statistical correlation in both spatial (two-dimensional) and temporal (one-dimensional) domains. Most image annotation approaches use full feature extraction, which has a clear defect: flattening data into vector form destroys relative positional relationships in spatial and temporal domains, potentially causing information loss and spatial orientation misjudgment of targets in images, while possibly introducing irrelevant information.

Temporal features can be extracted by computing element-wise products between frames. Using multiple parallel convolutional layers to extract features and computing pairwise element-wise products of these features enables multiplicative interactions between neurons, which can explicitly learn temporal dynamic spatial features while preserving CNN advantages in processing spatial features [12].

2 Model Construction

2.1 Overall Architecture of Attention-Based Encoder-Decoder

[Figure 3: see original paper] illustrates the overall architecture of the attention-based recurrent network encoder-decoder. The model first analyzes and represents multiple visual regions for image feature extraction, then employs an Attention LSTM structure (an LSTM network with attention mechanism in encoder-decoder) to predict sequences for different regions, finally generating visual attention-based annotation word sequences. This model can be viewed as a process of encoding high-dimensional raw input data and then decoding it into low-dimensional abstract features, processing associations between modules through the encoder-decoder framework [16].

LSTM is initialized with storage states and hidden states through average annotation vectors, obtained via two classification MLPs as shown in (9):

$$c_0 = f_{init,c}\left(\frac{1}{L}\sum_{i=1}^{L} a_i\right), \quad h_0 = f_{init,h}\left(\frac{1}{L}\sum_{i=1}^{L} a_i\right)$$

The attention function determines the attention quantity allocated to image feature a_i under hidden state h_{t-1}, where $\alpha_i \in [0,1]$ and $\sum_{i=1}^{L} \alpha_i = 1$. The probability of output vocabulary is jointly determined by image context vector c_t, previous word y_{t-1}, and hidden state h_t, as shown in (10), where θ represents learning parameters. Correspondingly, there is a loss function L_s for the negative sampling log probability of word s:

$$p(y_t|a, y_{t-1}) \propto \exp(W_o E y_{t-1} + U_o h_t + G_o c_t)$$

$$L_s = -\log p(w = s|a, y_{t-1})$$

2.2 Encoder: Convolutional Features

The model takes a single raw image and generates annotation words encoded from 1 to K, where K is vocabulary size and D is label length. Using CNN to extract annotation vectors $\{a_1, a_2, ..., a_L\}$ as feature vectors, the extractor produces L vectors corresponding to different spatial location features of the image, each represented by a D-dimensional vector.

To obtain correspondence between feature vectors and image parts, feature maps from convolutional layers are directly fed through fully connected layers to the next hidden layer containing 512 neural units. This enables the decoder to selectively focus on certain image parts and weight subsets of all feature vectors [17,18].

2.3 Decoder: Attention LSTM Network

Introducing visual attention mechanisms into the network allows each time step to adaptively concentrate on relatively small but information-rich image regions,

thereby accelerating model decoding. Using LSTM networks as decoders, the following formulas describe the LSTM unit operations:

$$
\begin{aligned}
i_t &= \sigma(W_i E y_{t-1} + U_i h_{t-1} + Z_i c_{t-1} + b_i) \\
f_t &= \sigma(W_f E y_{t-1} + U_f h_{t-1} + Z_f c_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_c E y_{t-1} + U_c h_{t-1} + Z_c c_{t-1} + b_c) \\
o_t &= \sigma(W_o E y_{t-1} + U_o h_{t-1} + Z_o c_t + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where i_t, f_t, c_t, o_t, h_t represent input gate, forget gate, memory cell, output gate, and hidden state at time t; W, U, Z, b are weight matrices and biases; E is the embedding matrix; and σ is the sigmoid function.

The context vector c_t is computed as a weighted sum of annotation vectors:

$$
c_t = \sum_{i=1}^{L} \alpha_{ti} a_i
$$

where α_{ti} is the attention weight for annotation a_i at time t, calculated by:

$$
\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}, \quad e_{ti} = f_{att}(h_{t-1}, a_i)
$$

2.4 Spatially-Guided Attention

Spatial factors [9] are crucial in image attention. As elaborated in Section 1.2, this paper incorporates spatial feature factors into the attention model to better achieve image annotation and generation. The final CNN layer (ResNet) has dimensions $2048 \times 7 \times 7$, where $A = \{a_1, a_2, ..., a_k\}$ represents fully connected layer spatial convolution features, with each grid position denoted by i. The global image feature is represented as:

$$
g = \frac{1}{k} \sum_{i=1}^{k} a_i
$$

where g denotes global image features. Using a single-layer perceptron and activation function to adjust image feature vectors, with weight parameters W and bias b, new feature vectors are obtained:

$$
v_i = \mathrm{ReLU}(W_a a_i + b_a), \quad v_g = \mathrm{ReLU}(W_g g + b_g)
$$

The final spatial image features are $V = \{v_1, v_2, ..., v_k\}$, each $v_i \in \mathbb{R}^d$ representing its corresponding image part. The spatial attention model computes the

LSTM context vector c_t as:

$$c_t = f_{att}(V, h_t) = \sum_{i=1}^{k} \alpha_{ti} v_i$$

where α_{ti} is the attention function, and spatial image features V and LSTM hidden state h_t are combined to predict the next word. The context vector c_t can serve as visual residual information for current hidden state h_t, reducing uncertainty in predicting the next word.

The attention distribution $\{\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tk}\}$ is obtained via a single-layer neural network with softmax function over the image:

$$\alpha_{ti} = \frac{\exp(z_{ti})}{\sum_{j=1}^{k} \exp(z_{tj})}, \quad z_{ti} = w^T \tanh(W_v v_i + W_g h_t)$$

where $w \in \mathbb{R}^d$, $W_v \in \mathbb{R}^{d \times d}$, $W_g \in \mathbb{R}^{d \times d}$, and $z_t \in \mathbb{R}^k$.

Interest attention is generated through model validation. Specifically, positive example validation annotation maps are provided by positive example validation annotations and can be considered as two attention probability distributions, typically used in cross-entropy loss verification. For words not aligned with image regions (e.g., “of” , “is”), setting $\beta_{ti} = 0$ makes the total loss a weighted sum of two loss terms:

$$L = \lambda L_{cap} + (1 - \lambda) L_{attn}$$

where L_{cap} is the captioning loss and L_{attn} is the attention loss.

3 Experiments

3.1 Experimental Setup and Evaluation Metrics

Experiments are conducted on two open-source datasets: Flickr30k and COCO. Flickr30k contains 31,783 images collected from Flickr, mostly depicting human daily activities with manual annotations of five sentences per image. COCO is currently the most widely used dataset for image captioning, containing 82,783 training images, 40,504 validation images, and 40,775 test images. Due to the excessive training time for all images, a subset is randomly sampled and combined as the experimental dataset: 4,000 training images, 500 validation images, and 500 test images, with each image having five manual annotations. Validation images are primarily used to determine model parameters and are subsequently merged into the training set after parameter determination [21,22].

The experimental platform is an HP desktop with a 3.2 GHz Intel i5 CPU, 4.0 GB RAM, Ubuntu 14.0 operating system, MATLAB 2014a, and Python 2.7. Common evaluation metrics for image caption generation include BLEU [25],

METEOR [23], and CIDEr [24]. This experiment employs these metrics for evaluation.

[Figure 4: see original paper] shows positive examples from the experiment. Based on the attention model' s image semantic generation annotation positive samples, the primary attention focuses on features of three people in the image, with darker colors indicating higher attention weights. Consequently, words like "man" and "boy" have slightly higher weights than others. Due to the incorporation of spatial features, pairwise human relationships can be inferred—simultaneous attention to regions suggests possible spousal relationships, while male attention on a boy suggests fatherhood. Annotated vocabulary predictions are derived from logical prediction results of existing statement vocabulary in the training set.

[Figure 5: see original paper] compares experimental results. With spatial factors incorporated, attention weights and spatial judgment are corrected. The figure shows attention comparisons for the same traffic sign under three existing models. The attention weight range for the STOP sign is smaller than the Hard Attention model. Due to spatial features removing irrelevant features and deepening relevant feature weights, it represents more typical full feature recognition than DeepVS, though with excessive color space recognition.

3.2 Experimental Results Analysis

All experimental parameters strictly follow Xu et al. [5]. Images are resized to 256 pixels on the short side and center-cropped to 224×224 pixels. After pretraining on ImageNet, conv5_4 features from VGG19 network are extracted, with the top convolutional layer sized 14×14. For visualizing attention model weights, the upsampling weight factor is $2^4 = 16$, using a Gaussian filter to simulate receptive field size. CNN convolution iterations are set to 15,000, and training text vector matrix iterations to 15,000. To avoid overfitting, CNN weight decay is set to 10^{-3}. The LSTM language model learning rate is 4×10^{-4}; update weight parameters are set to $\alpha = 0.8$, $\beta = 0.999$. Stochastic gradient descent without regularization is performed, with 1,300 LSTM units for Flickr30k and 1,800 for COCO datasets.

Visualization of vocabulary generation processes is shown in [Figure 6: see original paper]. Non-attention words like "of" receive enhanced attention focus, as articles are likely followed by key nouns. Words such as "riding" and "elephant" are allocated larger attention probability weights than non-attention words. Visual attention probability allocation for the same word differs across different contextual scenarios. For example, the word "a" typically has high annotation probability at sentence beginnings, as background context requires LSTM to preserve information for subsequent judgment.

Further visualization of annotation generation corresponding to spatial attention relationships is shown in [Figure 6: see original paper]. The context vector c_t combines with hidden state h_t to predict the next word y_{t+1}. Current hid-

den state h_t determines where to attend, combining two information sources to predict the next word. The generated context vector c_t serves as visual residual information for current hidden state h_t, reducing uncertainty in next-word prediction.

TABLE:1 compares attention models on Flickr30k and MS-COCO datasets, using M for METEOR, C for CIDEr, and Ours for our method. Compared to algorithms without spatial feature attention, our method slightly outperforms other attention models in local performance. On Flickr30k, CIDEr score improves from 0.248 to 0.255; on COCO, from 0.987 to 0.990. On COCO, BLEU-4 improves from 0.326 to 0.331, and METEOR from 0.250 to 0.257. The annotation model' s BLEU metrics improve over the baseline. On Flickr30k, BLEU-1 increases by $(0.671 - 0.668)/0.668 = 0.4\%$; on COCO, BLEU-2 increases by $(0.570 - 0.547)/0.547 = 4.2\%$. **TABLE:1** shows our method achieves good annotation performance in accuracy, with a 19.2% improvement on Flickr30k and 2.1% on MS-COCO through training and random local result sampling comparison.

Computational complexity comparison involves randomly sampling and testing 1,000 images from both datasets, comparing average annotation time per image for sequences within 20 characters. Results are shown in **TABLE:2**. Our algorithm' s runtime increases by 0.039–0.320 seconds compared to other models, with a relative increase of 0.11 on Flickr30k and 0.02 on MS-COCO, averaging 0.07 overall—well below 1, indicating acceptable complexity increase. Comprehensive scoring metrics demonstrate that our spatial attention fusion annotation model performs well locally and possesses practical value.

TABLE:2 Average Runtime on Datasets (seconds/image) | Model | Flickr30k | MS-COCO | |—-|——-|——| | DeepVS | 0.89 | 1.23 | | Hard-Attention | 0.92 | 1.28 | | MSM | 0.85 | 1.19 | | Ours | 0.99 | 1.25 |

4 Conclusion

Building upon previous work, this paper proposes an effective visual attention model incorporating spatial features for images, capable of well-describing attention-attracting regions. First, image features are obtained through convolutional neural networks, with feature image region annotation matching. Using an LSTM model with attention mechanism as encoder-decoder, image annotation is performed through attention weighting combined with spatial features, ultimately generating image annotation results based on spatial feature attention. Experimental results demonstrate that compared to related methods, the proposed algorithm achieves certain performance improvements in annotation. However, considerable improvement is still needed in overall evaluation and originality.

References

[1] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. Computer Science, 2014, 40 (12): 4751-4763.

[2] Kiros R, Salakhutdinov R, Zemel R. Multimodal neural language models [C]// Proc of International Conference on Learning Representations. 2014: II-595.

[3] Vinyals O, Toshev A, Bengio S, et al. Show and tell: a neural image caption generator [J]. Computer Science, 2015, 36 (7): 3156-3164.

[4] Karpathy A, Li F F. Deep visual-semantic alignments for generating image descriptions [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2014, 39 (4): 664-676.

[5] Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention [J]. Computer Science, 2015, 58 (12): 2048-2057.

[6] Yang Z, Yuan Y, Wu Y, et al. Review networks for caption generation [C]// Advances in Neural Information Processing Systems. 2016: 2361-2369.

[7] You Q, Jin H, Wang Z, et al. Image captioning with semantic attention [J]. Computer Science, 2016, 42 (13): 4651-4659.

[8] Wu Q, Shen C, Liu L, et al. What value do explicit high level concepts have in vision to language problems? [J]. Computer Science, 2016, 12 (01): 1640-1649.

[9] Lu J, Xiong C, Parikh D, et al. Knowing when to look: adaptive attention via a visual sentinel for image captioning [J]. International Journal of Computer Vision, 2016, 115 (3): 211-252.

[10] Zhou L, Xu C, Koch P, et al. Watch what you just said: image captioning with text-conditional attention [J]. IEEE Trans on Image Processing, 2016, 25 (8): 3919-3930.

[11] Zhang Chong. Research on text classification technology based on Attention-Based LSTM model [D]. Nanjing: Nanjing University, 2016.

[12] Yang Gelan, Deng Xiaojun, Liu Cong. Facial expression recognition model based on deep spatio-temporal convolutional neural network [J]. Journal of Central South University: Natural Science Edition, 2016, 47 (7): 2311-2319.

[13] Li Jing. Research on image annotation based on multiple features [D]. Wuhan: Wuhan University of Technology, 2013.

[14] Teng Fei, Zheng Chaomei, Li Wen. Long short-term memory multi-dimensional topic sentiment tendency analysis model [J]. Computer Applications, 2016, 36 (8): 2252-2256.

[15] Liu Jie. Implementation of LSTM neural network on Android platform [D]. Tianjin: Nankai University, 2016.

[16] Fu Kun, Jin Junqi, Cui Renpeng, et al. Aligning where to see and what to tell: image captioning with region-based Attention and scene-specific contexts [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2015, 39 (12): 2321-2334.

[17] Cho K, Merrienboer B V, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [J]. Computer Science, 2014, 45 (18): 4913-4921.

[18] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks [J]. Advances in Neural Information Processing Systems, 2014, 4 (3): 3104-3112.

[19] Mao J, Xu W, Yang Y, et al. Deep Captioning with multimodal recurrent neural networks (m-RNN) [C]// Proc of International Conference on Learning Representations. 2015: II-301.

[20] Liu C, Mao J, Sha F, et al. Attention correctness in neural image captioning [C]// Proc of AAAI - the Association for the Advance of Artificial Intelligence. 2017: 4176-4182.

[21] Ke Xiao, Li Shaozi, Cao Donglin. Research on automatic image annotation method based on relevant visual keywords [J]. Journal of Computer Research and Development, 2012, 49 (4): 846-855.

[22] Denkowski M, Lavie A. Meteor universal: language specific translation evaluation for any target language [C]// Proc of Workshop on Statistical Machine Translation. 2014: 376-380.

[23] Vedantam R, Zitnick C L, Parikh D. CIDEr: consensus-based image description evaluation [J]. Computer Science, 2014, 9 (4): 4566-4575.

[24] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation [C]// Proc of Meeting on Association for Computational Linguistics. 2002: 311-318.

[25] Yao T, Pan Y, Li Y, et al. Boosting image captioning with attributes [J]. ACM Trans on Graphics, 2016, 27 (3): 1423-1436.

[26] Zhang J, Lin Z, Brandt J, et al. Top-down neural attention by excitation backprop [C]// Proc of European Conference on Computer Vision. [S. l.]: Springer International Publishing, 2016: 543-559.

Note: Figure translations are in progress. See original paper for figures.