An Attention-based Regression Model for Grounding Textual Phrases in Images

Ko Endo and **Masaki Aono**
Toyohashi University of Technology
{k-endo@kde.cs.tut.ac.jp,aono@tut.jp}

Eric Nichols and **Kotaro Funakoshi**
Honda Research Institute Japan
{e.nichols,funakoshi}@jp.honda-ri.com

Abstract

Grounding, or localizing, a textual phrase in an image is a challenging problem that is integral to visual language understanding. Previous approaches to this task typically make use of candidate region proposals, where end performance depends on that of the region proposal method and additional computational costs are incurred. In this paper, we treat grounding as a regression problem and propose a method to directly identify the region referred to by a textual phrase, eliminating the need for external candidate region prediction. Our approach uses deep neural networks to combine image and text representations and refines the target region with attention models over both image subregions and words in the textual phrase. Despite the challenging nature of this task and sparsity of available data, in evaluation on the ReferIt dataset, our proposed method achieves a new state-of-the-art in performance of 37.26% accuracy, surpassing the previously reported best by over 5 percentage points. We find that combining image and text attention models and an image attention area-sensitive loss function contribute to substantial improvements.

1 Introduction

In recent years, semantic grounding has been an active area of research in artificial intelligence. In particular, progress has been made in multimodal tasks, such as image captioning and image QA, with deep learning [Xu *et al.*, 2015; Lu *et al.*, 2016]. Solving this problem is essential for computers to understand how humans communicate about what they see and has many applications, such as robot vision. In this paper, we focus on the task of grounding textual phrases in images.

Previous approaches treat grounding of textual phrases in images as a ranking problem, assigning scores to candidate image regions–which are extracted from the image in advance–based on their relevance to the textual phrase. Thus, previous methods are based on the premise that the correct region is included in the candidate regions, however, this is not always the case. In addition, previous methods scored candidate regions with the likelihood of generating the textual phrase from an image captioning model, leading to difficulty if the target region is not cleanly contained inside a candidate region proposal. To avoid these problems, we treat region prediction as a regression problem and propose a method to directly identify the region using deep learning with attention models. The objective variables of this regression are the top left x and y coordinates, width, and height of a region. Since our proposed method outputs the region itself, it does not require external candidate region proposals, unlike previous approaches. Recently, attention models have been shown to contribute to improved performance in various computer vision tasks [Xu *et al.*, 2015; Yang *et al.*, 2016; Lu *et al.*, 2016]. We apply attention models to learn fine-grained correspondences between text and image regions.

Our proposed method consists of five components, as shown in Figure 1. The first component is an image representation which is constructed by applying a convolutional neural network (CNN) to the target image. The second component is a text representation which is constructed by applying LSTMs to the textual phrase. The third component is an image attention model which estimates how much each region in an image is associated with the textual phrase. This image attention model makes predicting the target region's bounding box much easier. The fourth component is a text attention model that estimates the importance of each word based on the current image attention model's output. This model helps us refine the text representation to reflect the most important words. In the image grounding task, the most important information is *where*. Thus, as input to the text attention model, we use the image attention map, which represents where in the image is being attended to by the whole input phrase. The fifth component is a target region prediction model that performs regression over the image and text attention results to predict the region's bounding box in four parameters: the top left corner's x and y coordinates, width, and height of the bounding box. Our method thus learns to understand the image through the image model, the textual phrase through the text model, and the relationship between the image and textual phrase though the image and text attention models.

Our main contributions are as follows. First, we propose a regression model for identifying the region in an image corresponding to a textual phrase that does not rely on external image region predictions, eliminating a potential bottleneck in both accuracy and computational efficiency. Second, we show through detailed evaluation on the ReferIt

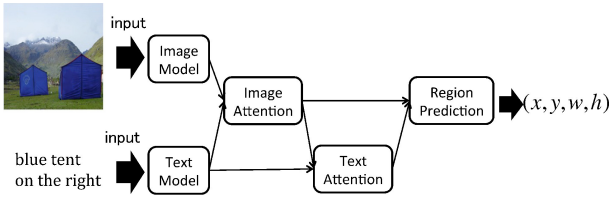


Figure 1: Our proposed method

dataset [Kazemzadeh *et al.*, 2014] that our proposed method achieves a new state-of-the-art of 37.26%, surpassing the previous best-performing method by over 5 percentage points. Third, we confirm that, by applying attention models over both image and text, our approach is able to learn fine-grained correspondences, further improving its performance.

2 Related Work

Hu et al. [2016b] proposed a Spatial Context Recurrent ConvNet (SCRC) to predict the region in an image corresponding to a given textual phrase. SCRC extracts candidate regions in advance using EdgeBox [Zitnick and Dollár, 2014], and then it assigns a score to each candidate region based on the probability of a given textual phrase being generated by an image captioning model from that candidate region. SCRC takes advantage of spatial information as well as image context features for candidate regions when computing the probability. Training SCRC requires full images, target image region annotations, and the associated textual phrase. However, it is difficult to collect large amounts of such data, so SCRC uses a pre-trained image captioning model to generate textual phrases from candidate regions.

To cope with the problem of sparse training data, Rohrbach et al. [2016] proposed a method called GroundeR that uses unsupervised learning. Like SCRC, GroundeR scores candidate regions with a caption generation model, but it also uses an attention model to decide which candidate region is most likely to be referred to by a textual phrase. It then generates the text of a gold standard caption, making it possible to learn image attention without correct region annotations. When the correct region is given, it is possible to supervise learning by directly fitting the predicted region to the correct region. Fukui et al. [2016] proposed Multimodal Compact Bilinear pooling (MCB) to extract multimodal feature. The outer product is more expressive than element-wise product and concatenation for making the multimodal feature. However, the outer product is typically infeasible due to its high dimensionality. MCB makes it possible to project the outer product to a lower dimensional space. They evaluate MCB on image QA and visual grounding. In the evaluation on visual grounding, their model receives the textual phrase and candidate regions as the input and makes multimodal feature from these. The output is the classification result that is which region is corresponds the textual phrase. Luo et al. [2017] proposed the model in which a generation model, which generates the expression for the region in the image, and a comprehension model, which selects the region corresponding to given expression, are integrated. First, the comprehension model is trained, and then the generation model is trained to generate the expression from which the comprehension model can select the correct region. Finally, Hu et al. [2016a] proposed a method which directly learns the semantic segmentation of the region corresponding to a textual phrase. Their method, however, requires gold standard semantic segmentation data, and they evaluate the Intersection-over-Union (IoU) between the correct region's segmentation and their predicted segmentation, making direct comparison to methods predicting bounding boxes infeasible.

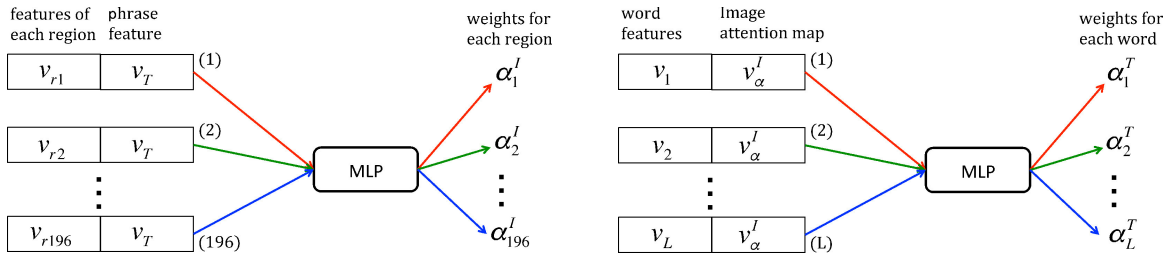
Our approach differs from [Hu *et al.*, 2016b; Rohrbach *et al.*, 2016; Fukui *et al.*, 2016; Luo and Shakhnarovich, 2017] by using regression to ground textual phrases, eliminating the need for external candidate regions and avoiding adverse influence from candidate region prediction quality. Unlike [Rohrbach *et al.*, 2016], it uses attention models over both image and text, learning more sophisticated relationships between image regions and text. Unlike [Hu *et al.*, 2016a], it does not require semantic segmentation annotations.

A task related to image region prediction is object detection, whose goal is to find a predetermined class of objects in image. Blaschko and Lampert [2008] proposed a method for detecting a single object in a single image using Support Vector Machines (SVM). Their proposed SVM learns a mapping of an input image to a bounding box. When training, they use Intersection over Union (IoU), which represents how much overlap occurs between the two regions, for the loss function. We designed our loss function with reference to this idea.

Bounding box regression has been used for refining candidate regions. Jaderberg et al. [2016] trained a CNN regression model that receives a candidate region as input and outputs a refined region. The primary difference between our work and theirs is that they use regression to refine candidate regions, while our approach uses it to directly predict a target region based on the results of attention models.

Recently, attention models have been applied in a variety of tasks, such as machine translation [Bahdanau *et al.*, 2015], textual question answering (QA) [Kumar *et al.*, 2015], and image caption generation [Xu *et al.*, 2015]. In the task of image QA, attention models have been used to detect and focus on the region in an image that is most related to answering a question. Attention in this situation makes it possible to boost the features in the relevant region, leading to more accurate answer prediction. Lu et al. [2016] proposed a method which combines both text and image attention at three different levels: word, phrase, and sentence. They introduced a function to specify which word or phrase in a question to attend to in an image, and performed a bidirectional attention from both question-to-image and image-to-question. Attention, thus, is effective in tasks requiring multimodal understanding.

Our approach is related to the attention models used in image QA. The textual phrase specifying an image region corresponds to a question, and predicting the target region corresponds to predicting the answer to the question. The difference between our approach and the approach for image QA is the target of the attention. In image QA, the attention model estimates the image region most related to the answer. In contrast, our image attention model directly predicts the region related to the textual phrase. Furthermore, attention-



(a) image attention (b) text attention

Figure 2: Overview of attention models. The image and the text attention models each have an independent MLP that is used to compute the weights associated with each target attention unit. Each target attention unit is paired with the input to that attention model and fed into the MLP one at a time. The red path in figures represents the first pair, the green path represents the second pair, and the blue path represents the last pair. In this way, both MLPs process the each feature pair independently.

based image QA methods use the features focused on by their attention models to produce a textual answer of some kind, whereas we use it to predict the region that the textual phrase refers to. Thus, in addition to image attention, we introduce a text attention model that estimates which words are important in the spirit of [Lu *et al.*, 2016].

3 Attention Models for Region Prediction

Here we describe our proposed method for predicting the bounding box of a region in an image given a textual phrase by regression over four variables (x, y, w, h), where (x, y) represents the left-top corner of the bounding box and (w, h) represents the width and the height of the bounding box. An overview of our proposed method is shown in Figure 1, consisting of an image representation generated with CNNs, a text representation generated with LSTMs, independent image and text attention models, and the region prediction regression model. In this section, we elaborate on these components and describe the loss function used to train the model.

3.1 Image Model

We use the CNN VGG 16-layer model of [Simonyan and Zisserman, 2014] to extract image features images. Specifically, we extract image features from $\boldsymbol{v}_I$, the output of intermediate layer conv5-3. Conv5-3 outputs 512 feature maps of size 14×14. Since the VGG 16-layer accept a 224×224 pixel image as input, one feature map produced by conv5-3 corresponds to a 16×16 region of the input image, amounting to a $14 \times 14 \times 512$ dimension feature vector. Hereafter, we denote this image feature by $\boldsymbol{V}_{map} = (\boldsymbol{v}_{r1}, \boldsymbol{v}_{r2}, ..., \boldsymbol{v}_{r196})$, where $\boldsymbol{v}_{ri} \in \mathbb{R}^{512}$ is the feature for region i.

3.2 Text Model

In order to construct feature representations from text, we use Bidirectional LSTMs (BLSTM) to extract textual phrase feature $\boldsymbol{v}_T$ and word features $\boldsymbol{v}_l$. The text attention model computes how much attention is given to each word feature outputted by the text model. BLSTMs are used to prevent the model from learning representations that are biased toward words near the end of the textual phrase. Given each word in the textual phrase represented by one-hot vector $\boldsymbol{q}_l$ whose size is the word embeddings' vocabulary size and the dimension associated with a target word set to 1 and all other dimension set to 0, first we transform each word into a distributed representation $\boldsymbol{u}_l$ denoted by embedding matrix $\boldsymbol{W}_e$,

$$\boldsymbol{u}_l = \boldsymbol{W}_e \boldsymbol{q}_l, \quad l \in 1, 2, ..., L$$

where l denotes the position where the word appears, and L denotes the length of the phrase. The BLSTM outputs are computed as follows:

$$\begin{aligned} \overrightarrow{\boldsymbol{v}_l} &= LSTM_{fwd}(\boldsymbol{u}_l) \\ \overleftarrow{\boldsymbol{v}_l} &= LSTM_{bwd}(\boldsymbol{u}_l) \\ \boldsymbol{v}_l &= \tanh(\boldsymbol{W}_{fwd}\overrightarrow{\boldsymbol{v}_l} + \boldsymbol{W}_{bwd}\overleftarrow{\boldsymbol{v}_l} + \boldsymbol{b}_{bi}) \end{aligned}$$

In the above equations, $LSTM_{fwd}$ denotes an LSTM encoding the textual phrase in the forward direction. $LSTM_{bwd}$ denotes an LSTM encoding the textual phrase in the backward direction. To merge the $\overrightarrow{\boldsymbol{v}_l}$ and $\overleftarrow{\boldsymbol{v}_l}$, we use single layer perceptron from the result of the preliminary experiment. The gates ($\boldsymbol{i}_l$,$\boldsymbol{f}_l$,$\boldsymbol{o}_l$,$\boldsymbol{z}_l$), memory cell ($\boldsymbol{c}_l$), and the hidden state ($\boldsymbol{v}_l$) of the LSTM are computed as follows:

$$\begin{aligned} \boldsymbol{i}_l &= \sigma(\boldsymbol{W}_{ui}\boldsymbol{u}_l + \boldsymbol{W}_{vi}\boldsymbol{v}_{l-1} + \boldsymbol{b}_i) \\ \boldsymbol{f}_l &= \sigma(\boldsymbol{W}_{uf}\boldsymbol{u}_l + \boldsymbol{W}_{vf}\boldsymbol{v}_{l-1} + \boldsymbol{b}_f) \\ \boldsymbol{o}_l &= \sigma(\boldsymbol{W}_{uo}\boldsymbol{u}_l + \boldsymbol{W}_{vo}\boldsymbol{v}_{l-1} + \boldsymbol{b}_o) \\ \boldsymbol{z}_l &= \tanh(\boldsymbol{W}_{cu}\boldsymbol{u}_l + \boldsymbol{W}_{cv}\boldsymbol{v}_{l-1} + \boldsymbol{b}_c) \\ \boldsymbol{c}_l &= \boldsymbol{f}_l\boldsymbol{c}_{l-1} + \boldsymbol{i}_l\boldsymbol{z}_l \\ \boldsymbol{v}_l &= \boldsymbol{o}_l \tanh(\boldsymbol{c}_l) \end{aligned}$$

where σ denotes the sigmoid activation function.

For the word features, we use $\boldsymbol{v}_l$, and for the phrase feature $\boldsymbol{v}_T$, an element-wise mean vector computed as follows:

$$\boldsymbol{v}_T = \text{mean}(\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_L)$$

Our system does not make use of any explicit semantic or syntactic information outside of what is captured from word context during the training of the embeddings.

3.3 Image Attention Model

Given the image feature maps $\boldsymbol{V}_{map}$ and the text representation $\boldsymbol{v}_T$, the image attention model estimates how much each region in the image corresponds to the textual phrase. As

Hyper-parameter	Final	Range
LSTM layers	1	[1,3]
Hidden size of image attention model	750	[100,800]
Layers of image attention model	3	[1,3]
Hidden size of text attention model	500	[100,500]
Layers of text attention model	1	[1,3]
Hidden size of region prediction model	450	[100,500]
Layers of region prediction model	2	[1,3]
Dropout	0.6	[0.5,0.8]
Batch size	20	-
Word embedding dimensions	300	-
Loss λ_1	0.0001	-
Loss λ_2	0.00001	-
Adam α	0.001	-
Adam β_1	0.9	-
Adam β_2	0.999	-

Table 1: The hyper-parameter search space and final values.

Method	Accuracy (%)
SCRC [Hu *et al.*, 2016b]	17.93
GroundeR [Rohrbach *et al.*, 2016]	26.93
MCB [Fukui *et al.*, 2016]	28.91
Comprehension [Luo and Shakhnarovich, 2017]	31.85
Our proposed method	**37.26**

Table 2: Comparison of our method to previous works.

shown in the following equations, we use a Multilayer Perceptron (MLP) to compute the weight α_i^I, which represents how much the model attends to region i, for each region.

$$\begin{aligned} \boldsymbol{h}_{vi} &= \tanh(\boldsymbol{W}_{vr}\boldsymbol{v}_{ri} + \boldsymbol{W}_{vt}\boldsymbol{v}_T + \boldsymbol{b}_v) \\ \alpha_i^I &= \mathrm{relu}(\boldsymbol{W}_{vh}\boldsymbol{h}_{vi} + \boldsymbol{b}_{vh}) \end{aligned}$$

where $\boldsymbol{h}_{vi} \in \mathbb{R}^n$ is the hidden state, $\boldsymbol{W}_{vh} \in \mathbb{R}^{1\times n}$, and $\boldsymbol{b}_{vh} \in \mathbb{R}^1$. Thus, α_i^I is a scalar. We repeat this process for the entire region, obtaining the image attention map $\boldsymbol{v}_\alpha^I = (\alpha_1^I, \alpha_2^I, ..., \alpha_{196}^I)$ in Figure 2 (a)[1]. The above equations are for a 3-layer MLP, however, the number of hidden layers and the size of hidden state n are tuned as hyper-parameters.

3.4 Text Attention Model

Given the word features $\boldsymbol{v}_l$ and the image attention map $\boldsymbol{v}_\alpha^I$, the text attention model estimates how important each word is and updates the phrase feature accordingly. As with the image attention, we use an MLP to compute the weight α_l^T for each word feature $\boldsymbol{v}_l$ as shown in Figure 2 (b).

$$\begin{aligned} \boldsymbol{h}_{tl} &= \tanh(\boldsymbol{W}_{tw}\boldsymbol{v}_l + \boldsymbol{W}_{t\alpha}\boldsymbol{v}_\alpha^I + \boldsymbol{b}_t) \\ \alpha_l^T &= \mathrm{softmax}(\boldsymbol{W}_{th}\boldsymbol{h}_{tl} + \boldsymbol{b}_{th}) \end{aligned}$$

where $\boldsymbol{h}_{tl} \in \mathbb{R}^m$ which is hidden state, $\boldsymbol{W}_{th} \in \mathbb{R}^{1\times m}$, $\boldsymbol{b}_{th} \in \mathbb{R}^1$. Similar to image attention, α_l^T is a scalar. Then, the phrase feature representation is updated as follows:

$$\boldsymbol{v}_T' = \sum_l^L \alpha_l^T \boldsymbol{v}_l$$

[1]It is typical to use softmax as the activation function for attention models, however, in preliminary experiments we experienced difficulties training, and switching to relu allowed it to succeed.

Method	Accuracy (%)
Full (IA, TA, penalty, GloVe 840B)	**37.26**
−Text attention	35.55
−Text attention and image attention	29.43
−Loss penalty $L_S(\theta)$	35.41

Table 3: Evaluation of various settings. IA is the image attention, TA is the text attention, and penalty means using $L_S(\theta)$ in the loss function. "−" means "without that setting."

Word embeddings	Accuracy (%)
GloVe 840B word embeddings	**37.26**
GloVe 42B word embeddings	36.60
GloVe 6B word embeddings	36.85
Random word embeddings	35.70

Table 4: Comparison between various word embeddings.

Again, the number of hidden layers and the size of hidden state m are hyper-parameters.

3.5 Region Prediction Model

We employ an MLP that takes the image attention map $\boldsymbol{v}_\alpha^I$ and updated phrase feature $\boldsymbol{v}_T'$ as input and predicts a bounding box specified by (x, y, w, h) as output. The hidden layer and the output layer are computed by the following formula:

$$\begin{aligned} \boldsymbol{h}_p &= \tanh(\boldsymbol{W}_{pv}\boldsymbol{v}_\alpha^I + \boldsymbol{W}_{pt}\boldsymbol{v}_T' + \boldsymbol{b}_{ph}) \\ \boldsymbol{g} &= \mathrm{relu}(\boldsymbol{W}_p\boldsymbol{h}_p + \boldsymbol{b}_p) \end{aligned}$$

where $\boldsymbol{g}$ is the bounding box. The number of hidden layers and the size of hidden state are tuned as a hyper-parameters, as with image and text attention.

3.6 Loss Function

We adopt Intersection over Union (IoU) [Everingham *et al.*, 2005] in order to define the error between the predicted region g and the ground truth region t. IoU represents the overlap between two regions, and is calculated as follows:

$$\mathrm{IoU}(t, g) = \frac{area(t \cap g)}{area(t \cup g)}$$

Blaschko et al [2008] defined the following Equation (1) as part of their loss functions to train SVMs for object detection.

$$L_{\mathrm{IoU}}(\theta) = 1 - \mathrm{IoU}(t, g) \quad (1)$$

We use this IoU loss to train our model. We also use the squared error between the ground truth and the predicted regions to directly optimize our model's output. The squared error is computed by:

$$L_{SE}(\theta) = \frac{1}{2}||\boldsymbol{t} - \boldsymbol{g}||^2$$

where we treat $\boldsymbol{t}$ and $\boldsymbol{g}$ as a 4-dimensional vector (x, y, h, w).

Furthermore, we introduce a penalty to help fit the image attention model area to the ground truth region. We define this penalty function $L_S(\theta)$ and the area of the image attention

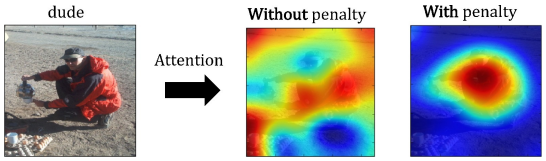


Figure 3: Visualization of the effect of loss penalty $L_S(\theta)$.

Rank	Error Cause	Count
1	Image attention	84
2	Incoherent bounding box prediction	55
3	Text attention	39
4	Attention requires inference	36
5	Target image region annotation	28
6	Textual phrase annotation	19

Table 5: Error analysis of 100 incorrect predictions.

S_{att} by the following equations:

$$L_S(\theta) = \frac{1}{2}(S_{box} - S_{att})^2$$
$$S_{att} = \sum_{i}^{196} \alpha_i^I$$

where S_{box} is the area of the ground truth in 14×14 feature maps.[2] The loss function $L(\theta)$ we propose is as follow:

$$L(\theta) = L_{\mathrm{IoU}}(\theta) + \lambda_1 L_{SE}(\theta) + \lambda_2 L_S(\theta)$$

where λ_1 and λ_2 are hyper-parameters.

4 Experiments

In this section, we describe the dataset we use, our model's configuration, experimental results, and discuss the advantages and disadvantages of our model.

4.1 Dataset

We used the ReferIt dataset [Kazemzadeh *et al.*, 2014] for both training and evaluation. This dataset consists of three parts: images, regions inside each image, and captions for each region. There are 20,000 total images, taken from the IAPRTC-12 dataset [Grubinger *et al.*, 2006]. The regions come from the SAIAPR-12 dataset [Escalante *et al.*, 2010]. There are approximately 120,000 total captions constructed in a two-player game with the goal of generating unambiguous referring expressions for target regions. There are approximately 100,000 total objects with 255 categories. These categories includes background such as sky. Therefore, this dataset includes bounding boxes of ambiguous size. We employed the same data splits as Hu et al. [2016b]: 9,000 images for training, 1,000 for validation, and 10,000 for testing to facilitate comparisons with prior approaches. In comparison, other multimodal datasets like MSCOCO image captioning [Lin *et al.*, 2014] and VisualQA [Antol *et al.*, 2015] have over 10 times as many images and 6-12 times as many unique image-annotation pairs as ReferIt[3], illustrating the data sparsity challenges faced in this task.

4.2 Model Configuration and Training

We performed hyper-parameter optimization using random search. The hyper-parameters and final settings are shown in Table 1. The size of each hidden layer is sampled by increment of 50. However, we selected the loss function weights from preliminary experiments because they have more influence over the training than other hyper-parameters. We fix the word embedding size to facilitate comparison between different embeddings and use the same 8,800 word vocabulary as [Hu *et al.*, 2016b]. We train the model with back propagation using Adam [Kingma and Ba, 2014] for SGD, with the authors' recommended values of hyper-parameters. We allow all parameters to be updated, except for the VGG 16-layer.

4.3 Results

In this section we present comparative evaluation against SCRC [Hu *et al.*, 2016b] and the current state-of-the-art method, GroundeR [Rohrbach *et al.*, 2016]. Following Hu et al. [2016b], we adopt accuracy as our evaluation criterion, where we consider a prediction correct if the overlapping IoU of the ground truth and the predicted area is $\geq 50\%$. Table 2 shows the comparison result with previous works. Our proposed method greatly outperforms previous works.

In order to confirm the effect of each component in our approach, we train models without text attention ($-$TA), without text and image attention ($-$TA and IA), and without the loss penalty ($-$loss penalty $L_S(\theta)$). Comparing the full model to them as shown in Table 3, we see that the full model outperforms the model without text attention by 1.71% points and the model without the text and image attention by 7.83% points. These result show that combining text and image attention is effective, but that image attention makes an especially large contribution. In addition, the model without text and image attention outperforms GroundeR by 2.50% points, showing that our regression approach is more effective than approaches that score candidate regions.

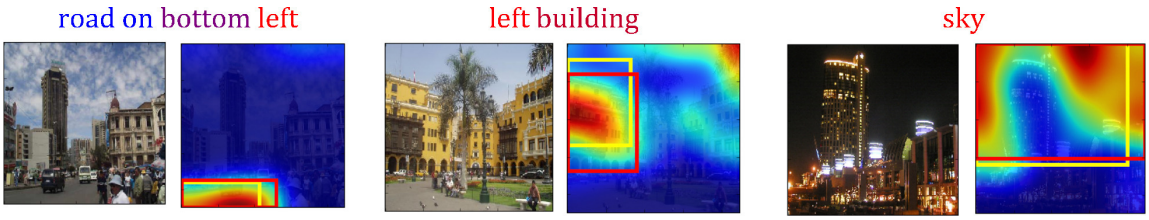
Furthermore, the full model outperforms the model without the loss penalty by 0.85% points, and the region corresponding textual phrase is more accurately focused as shown in Figure 3, showing that the loss penalty is effective.

We compare word embeddings as shown in Table 4. We evaluated on GloVe embeddings [Pennington *et al.*, 2014] because they have higher coverage than word2vec [Mikolov *et al.*, 2013] and released embeddings trained on various size datasets. We find that GloVe embeddings trained on the largest dataset of 840 billion words outperform the other GloVe embeddings by 0.41$-$0.66% points and randomly-initialized embeddings by 1.56% points[4], demonstrating the effectiveness of pre-trained word embeddings.

[2]It is possible that this penalty, when combined with the relu, which stands for the rectified linear unit activation function, in our image attention model, has an effect similar to normalization.

[3]MSCOCO has more than 300K images and 1.5M captions, and Visual QA has over 250K images with over 750K unique questions.

[4]We suspect GloVe 840B performed best because it had the largest vocabulary and training dataset of those tested.



(a) Correctly predicted examples

(b) Incorrectly predicted examples

Figure 4: Predicted examples. The input image is on the left, and the attention map is on the right. The red bounding box is the ground truth region, and the yellow bounding box is the predicted region. For visualization of attention, red represents high and blue represents low in both image and text attention.

4.4 Visualization of Attention

Examples of attention visualization are shown in Figure 4. The successful examples in Figure 4 (a) show that the region corresponding to the textual phrase is correctly focused on by the image attention. Furthermore, when a region in the background of the image is specified by the textual phrase, the image attention model successfully avoids focusing on objects in the foreground, as shown in the rightmost example. For text attention, words related to position such as *bottom* and *left* are given attention, demonstrating our model's ability to learn linguistic cues indicating location.

4.5 Error Analysis

We randomly sampled 100 erroneous predictions from our model and manually classified them by error causes. A single prediction can have multiple causes.

We find that the most common error cause is incorrect image attention, which occurs over 80% of the time. In most cases, the image attention is spread throughout the image and does not focus on the correct region as shown in the leftmost example in Figure 4 (b). In comparison, text attention errors, where a non-central expression is focused on, occur slightly less than 40% of the time. As can be seen by the text attention visualization in Figure 4 (b), despite using BLSTMs, text attention is often still biased toward the end of a sentence. Furthermore, in 36% of cases, a referring expression is attended more strongly than the object that is the target of the textual phrase. An example of this is *"2nd upright bike tire"* having weaker attention than *"closest to man"* in the middle example in Figure 4 (b). These errors show that our approach could likely benefit from hierarchical or cyclical attention models that allow for multiple passes over each attention module.

Gold annotation errors are another common cause of errors. A little less that 20% of the errors are caused by ambiguity, spelling errors, or unknown words in textual phrases, such as *womans* and *shelf?* in the leftmost example in Figure 4 (b). Furthermore, in 28% of cases, the ground truth regions given were ambiguous, either by being too big or small, or by not having well-defined boundaries, which is often the case for textual phrases such as *sky* or *corner*. These annotation errors may lead to errors in "image attention" and "text attention."

Finally, in over half of all errors, the bounding box predicted by our model does not capture a coherent region in the image, as shown in the rightmost example in Figure 4 (b), suggesting that despite our approach's independence from image region proposals, it may benefit from incorporating some kind of objectness judgement in its region predictions.

5 Conclusion

In this paper, we proposed a new attention-based method for directly predicting the region in an image specified by a textual phrase through regression. Through evaluation on the ReferIt dataset, we demonstrated that, despite the challenging nature of this task and sparsity of the dataset, our proposed method greatly outperformed the accuracy of all known methods by over 5% while eliminating the need for external image region candidates. Our evaluation and visualization of attention results also showed that the image and text attention and image attention size-based loss penalty greatly contribute to performance. Detailed analysis showed that attention failures and incoherent bounding box predictions were common causes of errors. In future work, we plan to refine our penalty function, incorporating measures of objectness. In addition, to better handle examples that require inference, we plan to explore attention architectures that allow multiple passes.

Acknowledgments

The part of this research is supported by MEXT KAKENHI, Grantin- Aid for Scientific Research (B), Grant Number 17H01746.

References

[Antol *et al.*, 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representationaa(ICLR)*, 2015.

[Blaschko and Lampert, 2008] Matthew B. Blaschko and Christoph H. Lampert. Learning to localize objects with structured output regression. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 2–15, Berlin, Heidelberg, 2008. Springer-Verlag.

[Escalante *et al.*, 2010] Hugo Jair Escalante, Carlos A. Hernández, Jesus A. Gonzalez, A. López-López, Manuel Montes, Eduardo F. Morales, L. Enrique Sucar, Luis Villaseñor, and Michael Grubinger. The segmented and annotated IAPR TC-12 benchmark. *Comput. Vis. Image Underst.*, 114(4):419–428, April 2010.

[Everingham *et al.*, 2005] Mark Everingham, Andrew Zisserman, Chris Williams, Luc Van Gool, and Moray Allan. The 2005 PASCAL visual object classes challenge. *First PASCAL Machine Learning Challenges Workshop (MLCW 05)*, pages 117–176, 2005.

[Fukui *et al.*, 2016] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv:1606.01847*, 2016.

[Grubinger *et al.*, 2006] M. Grubinger, P. Clough, H. Muller, and T. Deselaers. The IAPR TC-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, 2006.

[Hu *et al.*, 2016a] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[Hu *et al.*, 2016b] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[Jaderberg *et al.*, 2016] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal on Computer Vision(IJCV)*, 2016.

[Kazemzadeh *et al.*, 2014] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. ReferIt game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kumar *et al.*, 2015] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285, 2015.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[Lu *et al.*, 2016] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. *CoRR*, abs/1606.00061, 2016.

[Luo and Shakhnarovich, 2017] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. *arXiv:1071.03439*, 2017.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[Rohrbach *et al.*, 2016] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV2016*, 2016.

[Simonyan and Zisserman, 2014] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv.org*, 2014.

[Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[Yang *et al.*, 2016] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Zitnick and Dollár, 2014] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.