Visual Grounding via Accumulated Attention

Chaorui Deng[1*], Qi Wu[2*], Qingyao Wu[3†], Fuyuan Hu[4], Fan Lyu[4], Mingkui Tan[3†]
[1,3]School of Software Engineering, South China University of Technology, China
[2] Australia Centre for Robotic Vision, The University of Adelaide, Australia
[4] School of Electronic & Information Engineering, Suzhou University of Science and Technology, China
[1]secrdyz@mail.scut.edu.cn, [2]qi.wu01@adelaide.edu.au, [3]{qyw, mingkuitan}@scut.edu.cn

Abstract

Visual Grounding (VG) aims to locate the most relevant object or region in an image, based on a natural language query. The query can be a phrase, a sentence or even a multi-round dialogue. There are three main challenges in VG: 1) what is the main focus in a query; 2) how to understand an image; 3) how to locate an object. Most existing methods combine all the information curtly, which may suffer from the problem of information redundancy (i.e. ambiguous query, complicated image and a large number of objects). In this paper, we formulate these challenges as three attention problems and propose an accumulated attention (A-ATT) mechanism to reason among them jointly. Our A-ATT mechanism can circularly accumulate the attention for useful information in image, query, and objects, while the noises are ignored gradually. We evaluate the performance of A-ATT on four popular datasets (namely RefferCOCO, ReferCOCO+, ReferCOCOg, and Guesswhat?!), and the experimental results show the superiority of the proposed method in term of accuracy.

1. Introduction

Visual Grounding (VG) has attracted a lot of attention in recent years [9, 17, 28, 31, 32]. Unlike object detection which aims to detect the objects or the regions of interest given the pre-defined class labels, VG expects to understand the natural language query and then find out the target object of the query in the image. VG is an important technique for a machine to understand the real-world and communicate with a person as a human does. In particular, VG can be widely used in the visual understanding system and dialogue system of new generation intelligence devices.

VG requires a full understanding of both the image the natural language query. However, in real-world applications, the queries can be very complex and ambiguous. More critically, the scene in the image can be even harder to analyze due to its complicated structure. While a person can easily understand the query and locate the target object, it is difficult for a machine to effectively grasp the key point in the natural language and the visual content. To illustrate these challenges, we show a practical example in Figure 1, where the queries are from Guesswhat?! [3], ReferCOCO, ReferCOCO+, and ReferCOCOg [31]. In this example, the target of all queries is the woman in a black T-shirt, as located by the box. There are many irrelevant concepts in these queries, such as "man", "friends", "game", and the image depicts a complex scene with multiple instances. Thus, VG requires the machine to understand the complex reasoning in the query, as well as the spatial and semantic relationships among the instances in the image.

Query from GuessWhat?!:	**Query from ReferCOCO:**
is it a person? Yes is the person a man? No is it the left woman? No does she wear glasses? Yes	woman in black
	Query from ReferCOCO+:
	woman wearing glasses with smile
	Query from ReferCOCOg:
	a woman in black playing a game with her friends

Figure 1. In VG, the query can be a dialogue, a sentence or a phrase, and the target of the query is an instance in the image.

As a research direction across vision and language, VG has benefited from the development of Convolutional Neu-

*Equal contribution
†Corresponding author

ral Networks [7, 12], Recurrent Neural Networks [2, 6], and other areas such as Image Captioning [5, 26] and Visual Question Answering (VQA) [27]. However, the pioneers [4, 9, 17, 28, 31, 32] in VG simply combine all information curtly, where there may exist information redundancy and the latent relationships are not considered. For example, Hu *et al.* [9] employ three LSTMs to process linguistic information, local and global visual information separately. In [31], Yu *et al.* extract two types of features to encode the similarities and differences among all objects, but they still consider a general feature over all kinds of information. Recently, the attention mechanism has been widely used in many applications such as Natural Language Processing [2, 23], Image Captioning [29, 30] and VQA [4, 14, 16, 24] due to its effectiveness in dealing with information redundancies. However, in VG, few works have been studied with attention mechanism, except Rohrbach *et al.* [21] proposed an approach that ground the query by reconstructing a given phrase using a language attention mechanism. In this way, the redundancy in queries are reduced, but redundancies in images (such as irrelevant objects) still exist.

In this paper, we decompose the Visual Grounding problem into three sub-problems: 1) identify the main focus in the query; 2) understand the concepts in the image; 3) locate the most relevant object. We re-formulate these subproblems to three highly correlated attention problems, *i.e.*, 1) **which words to focus on in a query**; 2) **where to look in an image**; 3) **which object to locate**. Solving these three attention problems separately is a naive solution for the VG task. However, failing to consider the correlations among those attention problems, this approach may lead to a unsatisfactory performance. To this end, we further employ an accumulating process to combine all types of attention together and refine them circularly, where each type of attention will be utilized as a guidance when computing the other two. The proposed method, named as Accumulated Attention (A-ATT), is end-to-end, and is capable of handling different forms of natural language queries.

We evaluate the performance of A-ATT on four datasets with different forms of input query, *i.e.*, ReferCOCO and ReferCOCO+ with short phrases or full sentences, ReferCOCOg with long sentences and the GuessWhat?! with multiple rounds of dialogues. Experimental results show that our model outperforms the previous state-of-the-art on most splits of the datasets by a large margin. Moreover, our model can display the attended regions and highlighted words in the visual grounding process (see Figure 5), which increases the interpretability of the model.

2. Related Work

Visual Grounding, also known as Referring Expression Comprehension, requires a model to respond to a query by specifying a corresponding region in an image. Hu *et al.* [9] regard VG as a generalization of object detection and employ three LSTMs to process linguistic, local and global information, respectively. Rohrbach *et al.* [21] train a visual grounding model by reconstructing the query phrase using attention mechanism. In [17], Mao *et al.* propose the Maximum Mutual Information (MMI) method that considers whether a listener would interpret a referring expression unambiguously. Yu *et al.* [31] propose a visual comparative method (visdif) to discriminate the target object from the surrounding objects. In [32], a Speaker-Listener model is proposed, where the listener module can comprehend referring expressions and ground language. These pioneers provide good baselines and motivations for future works.

The neural attention mechanism has been widely used in different areas of computer vision and natural language processing, for example the attention models in image captioning[25, 29, 30], visual question answering (VQA) [14], machine translation [2] and machine reading systems [8]. In [2], Bahdanau *et al.* propose a "soft attention" mechanism which adds a layer to the network that predicts soft weights and uses them to compute a weighted combination of the items in memory. In [14], Lu *et al.* propose a hierarchical co-attention method for VQA which computes a conditional representation of the image given the question, as well as a conditional representation of the question given the image. Following [14], Wang *et al.* [24] extend the co-attention model to higher orders. However, in [14] and [24] the attention among each kind of information is computed only once, which may be insufficient to model the latent correlations effectively. Unlike these methods, the proposed accumulated attention mechanism attend on all kind of input information jointly for multiple times.

3. Proposed Method

The Visual Grounding task aims to ground a query into a region of an image. Formally, given an image I containing k objects $O = \{o_1, o_2, \ldots, o_k\}$ and a query Q, we hope to learn a hypothesis h to map Q to the target object o^*, *i.e.*, $h(I, O, Q) \rightarrow o^*$. In this paper, we propose to transform the VG problem into three attention problems: 1) **which words to focus on in a query**; 2) **where to look in an image**; 3) **which object to locate**. On the top of these three attention problems, we further propose a novel model that accumulate attention among all sources of information (I, O and Q) in a circular manner. Moreover, each type of attention can be refined by the other two during the accumulating process, and the model can be trained end-to-end.

In the following, we first introduce our attention modules in Section 3.1. In Section 3.2, we describe how to reason multiple kinds of attention jointly using the accumulated attention (A-ATT) mechanism. Lastly, we illustrate how to ground the query in the image with the proposed method.

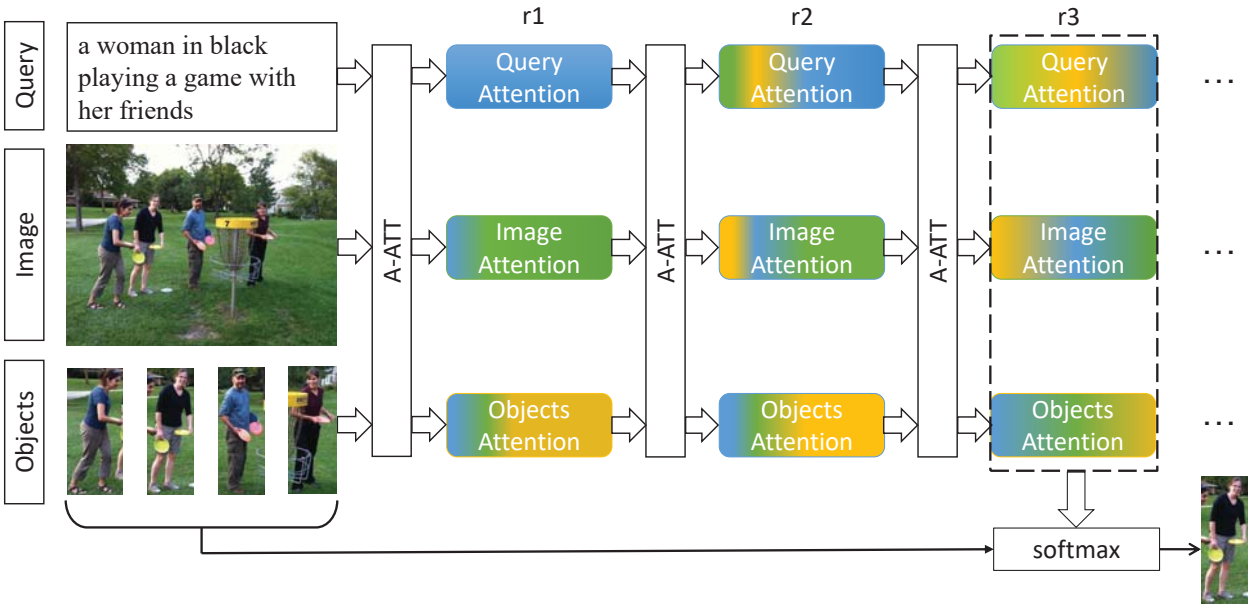


Figure 2. The architecture of the proposed method. Given an input image, the corresponding query and the object candidates, our model extract attentions from these three kinds of information. We use blue, green and yellow to represent the attention for the query, image, and objects, respectively. Performing the A-ATT mechanism for more rounds can facilitate the communication among different information. During each round of the proposed A-ATT mechanism, each type of attention will be refined by the other two.

3.1. Attention modules

3.1.1 Which words to focus on in a query

In the Visual Grounding task, a query can be a short phrase, a sentence or a multi-round dialogue, as shown in Figure 1. Here, we take the dialogues in GuessWhat?! [3] as a example. A dialogue is consisted of a sequence of question-answer (QA) pairs $Q = \{\{q_1, a_1\}, \{q_2, a_2\}, ..., \{q_T, a_T\}\}$, where T is the number of the QA pairs.

In general, dialogues can be very long, therefore we use a hierarchical architecture [14] to encode them. Specifically, for dialogue Q, we first use a word embedding layer to encode each word in the questions into a fix-length vector. Then, for each QA pair, we feed the encoded question into an LSTM and obtain the question feature from the hidden state at the last time step. Meanwhile, the answer feature is simply the one-hot encoding of the original answer. We then concatenate the question feature and answer feature together to obtain the feature of the QA pair. Lastly, we employ another LSTM which takes all QA pair features as the inputs, and collect the outputs at every recurrent step as the dialogue feature. Note that when the query is a short phrase or a sentence, a vanilla LSTM would be sufficient to capture the relationships within the sequence, thus utilizing a hierarchical architecture in this case is unnecessary and even results in over-fitting.

We denote the query feature as $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_T\}$, where $\boldsymbol{s}_i$ is the feature for i-th QA pair (or word) in the query. To explore which QA pairs in a dialogue to focus on, we use following attention mechanism:

$$\alpha_i^q = \text{softmax}(\boldsymbol{w}_q^\top \boldsymbol{H}_i^q), \quad i = 1, ..., T. \tag{1}$$

Here, $\boldsymbol{H}_i^q$ is a joint feature generated by an accumulate operation (see Equation (4)) from the query feature $\boldsymbol{S}$, where the features extracted from the other two kinds of information, *i.e.*, image and object, are utilized as guidance. We will explain the details of the calculation of $\boldsymbol{H}_i^q$ in Section 3.2. In addition, $\boldsymbol{w}_q$ is a learnable transformation matrix, and the resultant α_i^q is the attention weight for the i-th QA pair in the query.

3.1.2 Where to look in an image

To decide where to look in an image, we adopt the usual practice in Image Captioning [29] and VQA [14] that divide the image into multiple regions corresponding to the extracted feature map, and assign different attention weights for those regions. In our model, we employ the VGG-16 [22] model to extract feature maps $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_L\}$ from the last convolutional layer, where $L = 14 \times 14$ is the number of regions. Similar to the calculation of the query attention, we obtain a joint feature $\boldsymbol{H}^c$ from $\boldsymbol{V}$ to guide the attention among all the image regions:

$$\alpha_i^c = \text{softmax}(\boldsymbol{w}_c^\top \boldsymbol{H}_i^c), \quad i = 1, ..., L, \tag{2}$$

where the α_i^c is the attention weights for the i-th region in the image and $\boldsymbol{w}_c$ is the learnable parameters.

3.1.3 Which object to locate

In the common scenario, there may exist multiple object candidates within an image. In this section, we explicitly pay different attention to those object candidates with the guidance from the query and the global image context. This can be beneficial for the final grounding task, because the objects with higher attention weights will play a more important role in the subsequent attention process. The object candidates can be obtained through a pre-trained object detection model like Faster-RCNN [20], or a proposal generating model such as Edgebox [33], Objectness [1] and so on. For fair comparisons, we follow [31, 32] and use the object candidates provided by datasets.

Afterwards, we represent each candidate as a fix-length vector composed of two types of information: the spatial information and the local information. Following [3], we obtain the spatial information of each object candidate by:

$$\boldsymbol{o}^s = [\frac{x_{min}}{w_{img}}, \frac{y_{min}}{h_{img}}, \frac{x_{max}}{w_{img}}, \frac{y_{max}}{h_{img}}, \frac{x_{center}}{w_{img}}, \frac{y_{center}}{h_{img}}, \frac{w_{box}}{w_{img}}, \frac{h_{box}}{h_{img}}],$$

where w and h represent width and height, respectively. The subscript box denotes the bounding box, and subscript img stands for the image. For local information, we crop the image with the bounding box of each candidate and feed the cropped area into the aforementioned VGG-16 model. Then, we take the output at the second fully-connected layer as the local information $\boldsymbol{o}^l$. More importantly, we embed $\boldsymbol{o}^s$ and $\boldsymbol{o}^l$ into the same vector space with another two embedding matrices, as we expect that the two components of the candidate representation should have equal contributions. We further utilize a Batch Normalization [10] layer after each embedding matrix to scale the output. Finally, the resultant $\boldsymbol{o}^s$ and $\boldsymbol{o}^l$ are concatenated together to form the object candidate representation $\boldsymbol{o} = [\boldsymbol{o}^l, \boldsymbol{o}^s]$.

We represent all the object candidates in an image by $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_K\}$, where K is the number of the object candidates. The attention weight for $\boldsymbol{o_i}$ is then calculated based on the joint feature $\boldsymbol{H}^o$:

$$\alpha_i^o = \text{softmax}(\boldsymbol{w}_o^\top \boldsymbol{H}_i^o), \quad i = 1, ..., K, \tag{3}$$

where α_i^o is the attention weights for the i-th object candidate that will determine which object to locate, and $\boldsymbol{w}_o$ is a transformation matrix.

3.2. Accumulated Attention (A-ATT) model

So far we have proposed three attention modules for solving the three sub-problems. However, since those sub-problems are highly correlated, we still need a "core" to combine multiple types of information together, and generate a joint feature $\boldsymbol{H}$ to guide the assignment of attention.

	$\boldsymbol{X}$	$\boldsymbol{g_1}$	$\boldsymbol{g_2}$	$\tilde{\boldsymbol{x}}$
Round 1	$\boldsymbol{S}$	$\boldsymbol{0}$	$\boldsymbol{0}$	$\tilde{\boldsymbol{s}}_1$
	$\boldsymbol{V}$	$\tilde{\boldsymbol{s}}_1$	$\boldsymbol{0}$	$\tilde{\boldsymbol{v}}_1$
	$\boldsymbol{O}$	$\tilde{\boldsymbol{v}}_1$	$\tilde{\boldsymbol{s}}_1$	$\tilde{\boldsymbol{o}}_1$
Round 2	$\boldsymbol{S}$	$\tilde{\boldsymbol{o}}_1$	$\tilde{\boldsymbol{v}}_1$	$\tilde{\boldsymbol{s}}_2$
	$\boldsymbol{V}$	$\tilde{\boldsymbol{s}}_2$	$\tilde{\boldsymbol{o}}_1$	$\tilde{\boldsymbol{v}}_2$
	$\boldsymbol{O}$	$\tilde{\boldsymbol{v}}_2$	$\tilde{\boldsymbol{s}}_2$	$\tilde{\boldsymbol{o}}_2$
...	...	...	...	...

Figure 3. The A-ATT process. $\boldsymbol{X}$ denotes the input features, while $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are the attention guidances. $\tilde{\boldsymbol{x}}_i$ indicates the attended feature of $\boldsymbol{X}$ at the i-th round. "**0**" means that the attention guidance is not generated yet.

In this section, we describe how to obtain this $\boldsymbol{H}$ via the proposed accumulated attention mechanism.

Formally, we denote the input of an attention module as a feature sequence: $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \in \mathbb{R}^{d \times n}$, where d is the dimension of features, and n is the sequence length. Here, $\boldsymbol{X}$ can be the query feature ($\boldsymbol{S}$), the image feature ($\boldsymbol{V}$) or the feature of the object candidates ($\boldsymbol{O}$). Then, the joint feature can be calculated through:

$$\boldsymbol{H}_i = \tanh(\boldsymbol{W}_x \boldsymbol{x}_i + \boldsymbol{W}_{g_1} \boldsymbol{g}_1 + \boldsymbol{W}_{g_2} \boldsymbol{g}_2), \tag{4}$$

where $\boldsymbol{g}_1, \boldsymbol{g}_2 \in \mathbb{R}^d$ denote the attention guidance from the other two types of feature. $\boldsymbol{W}_x$, $\boldsymbol{W}_{g_1}$ and $\boldsymbol{W}_{g_2}$ are learnable parameters. Afterwards, we compute the attention weight α_i as in Equation (1), (2) or (3) for each element $\boldsymbol{x}_i$ in the feature sequence $\boldsymbol{X}$, and obtain the attended feature by:

$$\tilde{\boldsymbol{x}} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i, \quad \alpha_i \in \{\alpha_i^q, \alpha_i^c, \alpha_i^o\} \tag{5}$$

The attended feature $\tilde{\boldsymbol{x}}$ is then serving as an attention guidance $\boldsymbol{g}$ in other attention modules. Denoting the process in Equation (5) as $\tilde{\boldsymbol{x}} = \text{ATT}(\boldsymbol{X}, \boldsymbol{g}_1, \boldsymbol{g}_2)$, the attended features for all three kinds of information can be represented by:

$$\begin{cases} \text{ATT}(\boldsymbol{S}, \tilde{\boldsymbol{o}}, \tilde{\boldsymbol{v}}) = \tilde{\boldsymbol{s}} \\ \text{ATT}(\boldsymbol{V}, \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{o}}) = \tilde{\boldsymbol{v}} \\ \text{ATT}(\boldsymbol{O}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}}) = \tilde{\boldsymbol{o}} \end{cases} \tag{6}$$

Obviously, the computations in Equation (6) form a circulation: any type of attended feature will be reused to refine the attention of the other two kinds of information. We define Equation (6) as A-ATT, then we have:

$$\text{A-ATT}_{i+1} = \mathcal{R}(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{O}; \text{A-ATT}_i), \; i \in \{1, 2, \dots\}, \tag{7}$$

where $\mathcal{R}$ denotes the computations within a round and i indicates the i-th round of the circulation.

We illustrate the details of A-ATT process in Figure 3. Within each round of A-ATT, the features of query, image, and objects are fed into the ATT process progressively. Note

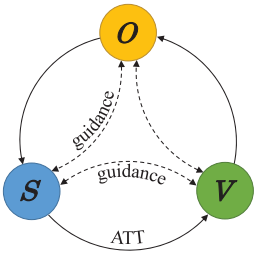


Figure 4. The A-ATT mechanism. Bold lines denote the ATT processes, and dash lines denote the attention guidance. One cycle denotes one round of A-ATT process.

that the first round of A-ATT is the initial phase, where no guidance information is available. At the end of each round, the attended features $\tilde{\boldsymbol{s}}$, $\tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{o}}$ will be generated through Equation (6), and passed to the next round through Equation (7). Moreover, as depicted in Figure 4, these attended features will keep flowing through the following rounds. During this "**attention accumulating**" process, the attention for the useful information in $\boldsymbol{S}$, $\boldsymbol{V}$ and $\boldsymbol{O}$ will accumulate, while the attention for the noises will vanish. Though the number of rounds can grow endlessly, the total model parameters are consistent throughout the process.

Dataset	# images	# queries	# objects
ReferCOCO	19,994	142,209	50,000
ReferCOCO+	19,992	141,564	49,856
ReferCOCOg	26,711	85,474	54,822
GuessWhat?!	66,537	155,280	134,073

Table 1. Data statistics

3.3. Grounding the query

To ground the query in the image, we first acquire the attended features of the query $\tilde{\boldsymbol{s}}$ and the image $\tilde{\boldsymbol{v}}$ at every rounds of the A-ATT process, then feed them into an embedding layer followed by a $\tanh(\cdot)$ activation function to generate a joint representation of the attended features, *i.e.*

$$\boldsymbol{F} = \tanh(\boldsymbol{W}_e[\tilde{\boldsymbol{s}}; \tilde{\boldsymbol{v}}]), \tag{8}$$

where $\boldsymbol{W}_e$ is an embedding matrix and $[\cdot]$ indicates the concatenation operation. Then, the probability for object $\boldsymbol{o}_i$ being the target object is computed by a softmax function:

$$p_i = \text{softmax}(\boldsymbol{F} \odot \tanh(\alpha_i^o \boldsymbol{W} \boldsymbol{o}_i)), \tag{9}$$

where $\boldsymbol{W}$ is another embedding matrix, and α_i^o is the attention weight for $\boldsymbol{o}_i$. Moreover, $\odot$ denotes element-wise multiplication, for we wish to explicitly find the best match between the fused image-query information $\boldsymbol{F}$ and the feature of object candidate $\boldsymbol{o}_i$. Let p^* be the predicted probability (softmax score) of the target object $\boldsymbol{o}^*$, our objective is to minimize the loss function: $\mathcal{L} = -\frac{1}{m}\sum_{j=1}^{m} \log p_j^*$, where m is the number of samples in a mini-batch.

4. Experiments

In this section, we first evaluate the proposed method on four popular datasets for visual grounding tasks, then we conduct several ablation studies on the effectiveness of each attention module and the benefit of the interactions among these attention modules. Specifically, we implement four versions of algorithms by performing the A-ATT process with one to four rounds, namely **Ours-r1**, **Ours-r2**, **Ours-r3** and **Ours-r4**. Other options (number of rounds larger than four) are not considered, because excessive rounds of A-ATT may lead to a heavy computational load.

4.1. Datasets

We evaluate the proposed method on four datasets based on MS-COCO [13], *i.e.* ReferCOCO, RegerCOCO+, ReferCOCOg and Guesswhat?!. In ReferCOCO and ReferCOCO+ [11], the queries are mostly short phrases, the difference is that the query in ReferCOCO+ should not contain any spatial information. Whilst in ReferCOCOg, the queries are normally declarative sentences. GuessWhat?! [3] is collected by a two-player game, the queries of which are all multi-round dialogues. The data statistics are recorded in Table 1.

Similar to [31], we split ReferCOCO and ReferCOCO+ into 40,000 training, 5,000 validation, and 5,000 testing samples, where the testing set are further split into "TestA" and "TestB". More precisely, images containing multiple people are put into "TestA" while images containing other objects are in "TestB". ReferCOCOg is split into 44,822 training and 5,000 validation samples. Following [3], we split the GuessWhat?! dataset into training, validation, and testing set by a fixed proportion of 70%, 15% and 15%.

4.2. Implementation details

In this paper, we use VGG-16 [22] as our backbone CNN to extract features from the whole image and the cropped regions. The VGG-16 model is pre-trained on ImageNet, and is fixed in our implementation. When processing queries, we use two types of recurrent architectures. Specifically, for a multi-round dialogue, we use the hierarchical LSTM [14] to process the dialogue at both QA pair level and dialogue level. For sentences or phrases, we simply take the vanilla LSTM to encode them. The size of word embedding is set to 256, while the dimensions of LSTM hidden states are all 512. The model parameters are optimized by momentum SGD, where the momentum coefficient is set to 0.9 and the learning rate is set to 0.1. We train the models for 20 epochs, and apply early stopping on the validation set.

4.3. Performance on ReferCOCO & ReferCOCO+ & ReferCOCOg

We first evaluate our methods on ReferCOCO, ReferCOCO+, and ReferCOCOg, where the queries are sen-

Methods	ReferCOCO			ReferCOCO+			ReferCOCOg
	Val acc	TestA acc	TestB acc	Val acc	TestA acc	TestB acc	Val acc
Baseline [17]	-	63.15	64.21	-	48.73	42.13	55.16
visdif [31]	-	67.57	71.19	-	52.44	47.51	59.25
MMI [17]	-	71.72	71.09	-	58.42	51.23	62.14
visdif+MMI [31]	-	73.98	76.59	-	59.17	55.62	64.02
Luo *et al.* [15]	-	74.14	71.46	-	59.87	54.35	63.39
Luo *et al.* (w2v)[15]	-	74.04	73.43	-	60.26	55.03	65.36
Neg Bag [19]	76.90	75.60	78.00	-	-	-	68.40
speaker+listener [32]	77.84	77.50	79.31	60.97	62.85	58.58	72.75
speaker+listener+reinforcer [32]	78.14	76.91	**80.10**	61.34	63.34	58.42	71.72
speaker+listener+MMI [32]	78.42	78.45	79.94	61.48	62.14	58.91	72.13
speaker+listener+reinforcer+MMI [32]	78.36	77.97	79.86	61.33	63.10	58.19	72.02
Ours-r1	79.19	79.67	78.16	64.45	66.51	58.84	72.33
Ours-r2	80.68	81.37	79.79	65.35	68.36	60.19	72.67
Ours-r3	80.98	**81.67**	79.96	65.50	67.92	**60.69**	72.94
Ours-r4	**81.27**	81.17	80.01	**65.56**	**68.76**	60.63	**73.18**

Table 2. Comparisons on ReferCOCO, ReferCOCO+ and ReferCOCOg.

tences or short phrases. We adopt Maximum Mutual Information method (MMI) [17], visdif method [31] and the listener model in speaker-listener architecture [32] as our main baselines. The comparative results are recorded in Table 2. On ReferCOCO dataset, **Ours-r3** achieves an accuracy of 81.67% on TestA split while the previous best result is 78.45% (in speaker+listener+MMI model). On TestB split, **Ours-r4** yields comparable performance to the state-of-the-arts (80.01% *vs*. 80.10%). Furthermore, on ReferCOCO+ dataset, all four versions of the proposed A-ATT mechanism consistantly improve the performance by a large margin on both TestA split(68.76%, **Ours-r4**) and TestB split(60.69%, **Ours-r3**). Finally, on ReferCOCOg dataset, A-ATT still surpasses the previous state-of-the-art methods (73.18% *vs*. 72.75%). These experimental results verify the superiority of our proposed A-ATT mechanism. Meanwhile, among all four versions of our methods, **Ours-r4** and **Ours-r3** generally outperform **Ours-r2** and **Ours-r1**, suggesting that A-ATT with more rounds of execution can generate better attended features for the visual grounding task.

Moreover, in the first round of A-ATT (*i.e*., **Ours-r1**), the accumulation process has not started, and **Ours-r1** can only achieve a slight or even no performance improvement among all the comparisons. Nevertheless, comparing **Ours-r2** with **Ours-r1**, we find that once the accumulation process is started, the performance improves significantly and immediately (*e.g*., 81.37% *vs*. 79.67% on TestA split, ReferCOCO), which clearly demonstrate the effectiveness of the accumulation process. It seems that more rounds (≥ 3) of A-ATT only yields slight improvement on the top of **Ours-r2**. One possible reason is that in VG tasks two or three rounds are already sufficient for the proposed A-ATT to achieve good interactions among all types of information, leaving the following rounds a small room to improve.

Model	Train err	Val err	Test err
Human	9.0%	9.2%	9.2%
Random	82.9%	82.9%	82.9%
LSTM	27.9%	37.9%	38.7%
HRED [3]	32.6%	38.2%	39.0%
LSTM+VGG [3]	**26.1**%	38.5%	39.2%
HRED+VGG [3]	27.4%	38.4%	39.6%
Ours-r2	29.3%	35.7%	36.5%
Ours-r3	30.5%	35.1%	35.8%
Ours-r4	29.8%	35.3%	36.3%
Ours-r3(w2v)	26.7%	**33.7**%	**34.2**%

Table 3. Comparisons on GuessWhat?!

4.4. Performance on GuessWhat?!

We then evaluate our model on GuessWhat?! [3], the queries in which are all multi-round dialogues. We follow [3] to take the bounding boxes and the corresponding categories to represent the object candidates. Table 3 shows the evaluation results. Our method outperforms the previous state-of-the-arts on both the validation and test split by a large margin, for example, **Ours-r2** outperforms the previous best result **LSTM** by 2.2%. With another round of the proposed A-ATT mechanism, we find that our model **Ours-r3** performs even better. However, **Ours-r4** produces a slightly worse result, which may due to the overfitting issues. Also note that some of the proposed models have higher training error than a few baseline models (*e.g*. **LSTM**, **HERD+VGG**, *etc*.), indicating that our proposed method is less likely to get overfitting. We additionally use a pre-trained word2vector [18] embedding on our best model Ours-r3, *i.e*., **Ours-r3(w2v)**, we observe that it can significantly boost the performance of the A-ATT based models, due to the better feature extracted from the queries.

4.5. Visualization of the attention

We further visualize the attention weights of the query, image and object candidates for RefCOCO, RefCOCO+, and RefCOCOg, as shown in Figure 5. We observe that the attention for different types of information tends to focus on the instances that are correlated semantically or spatially. For example, on the RefCOCO dataset, in the first column of the visualization results, the most focused word in the query is "person", while in the image the regions corresponding to the "person" are assigned larger weights. Meanwhile, among all the object candidates, the man sitting in front of the white box is attended. This means that different types of information can provide useful guidance for each other during the proposed A-ATT process.

Moreover, to illustrate the effect of the attention accumulating, we visualize the attention on RefCOCOg at different rounds of the A-ATT process. The results are shown in Figure 6. Obviously, from the second round (**r2**) to the fourth round (**r4**), the attention for all kinds of information (query, image and object candidates) become more concentrated on the relevant instances (*i.e.*, the target object, corresponding regions in the image, and keywords in the query), which means that attention weights for the target instances will accumulate through the A-ATT process.

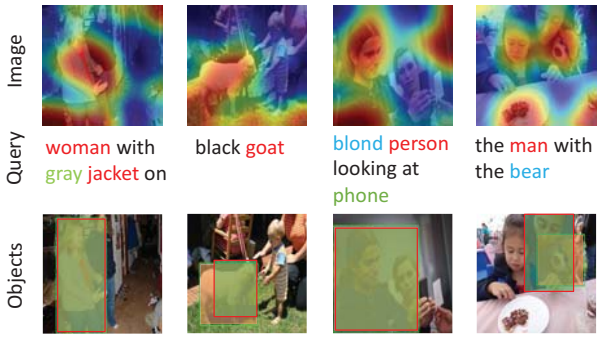
4.6. Ablation studies

In this section, we conduct several ablation studies by ignoring one or two types of attention in **Ours-r2** algorithm, and show the results in Table 4. In Table 4, **w/o. S**, **w/o. V** and **w/o. O** denote the models without the attention on queries, image and objects, respectively. **w/o. SO** denotes the model without attention on both queries and objects, **w/o. VO** and **w/o. SV** follow the same way. From Table 4 and Table 2, we can observe that the **Ours-r2** model outperforms all the ablation models, indicating that all types of attention information are necessary to the performance of the proposed A-ATT process. Meanwhile, we find that when using only one kind of attention information, (*i.e.*, there is no interaction between the attention modules, such as **w/o. SO**, **w/o. VO** and **w/o. SV**), the model can only yield comparable results to the baseline models. However, when at least two source of information are considered (such as **w/o. S**, **w/o. V**, **Ours-r2**, *etc.*), these models can significantly outperform the baseline models. This demonstrates that building interactions among the three attention modules is

	Models	w/o. S	w/o. V	w/o. O	w/o. SO	w/o. VO	w/o. SV
RefCOCO	Val acc	78.97	79.15	79.52	78.88	78.04	77.72
	TestA acc	79.14	79.82	80.08	78.75	78.43	78.09
	TestB acc	76.81	76.96	77.94	76.74	76.12	76.69
RefCOCO+	Val acc	63.06	63.33	64.22	62.44	62.63	62.73
	TestA acc	64.98	65.37	66.84	64.39	65.08	64.24
	TestB acc	58.70	58.23	59.08	58.24	57.36	58.60

Table 4. The results of ablation studies.

(a) Attention on ReferCOCO



(b) Attention on ReferCOCO+

(c) Attention on ReferCOCOg

Figure 5. Visualization of attention at the fourth round of A-ATT on ReferCOCO, ReferCOCO+, and ReferCOCOg. For image attention, we use colors from red to blue to represent the attention weights from large to small. For query attention, we use red, blue, green and black to indicate the attention weights on words that are high, above-average, below-average and negligible, respectively. For objects attention, we use the red mask to represent the candidates with the highest attention weight, and use the green mask to represent the ground-truth target object.

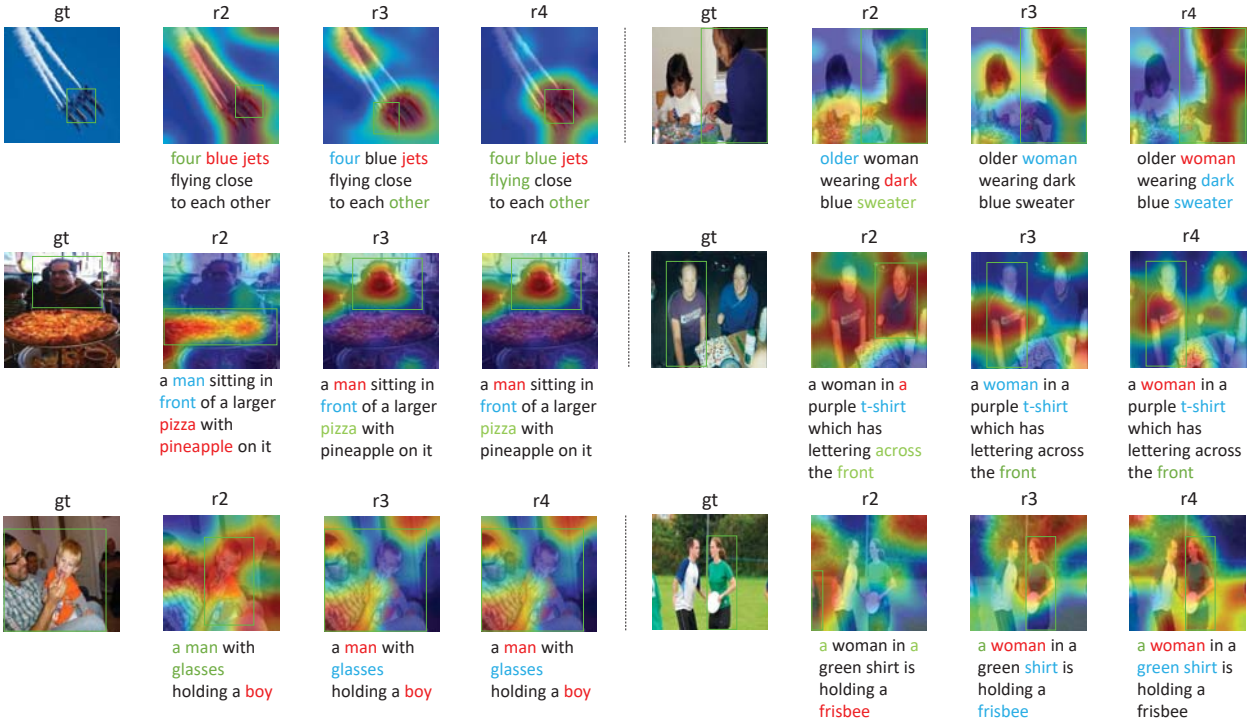


Figure 6. The process of attention accumulating. We visualize the attention for the query, image and objects at the second (r2), third (r3) and fourth (r4) round of the A-ATT process. The target objects are shown in the left (gt).

beneficial for the proposed A-ATT mechanism to achieve the state-of-the-art performance. Moreover, compared with **w/o. S** and **w/o. V**, **w/o. O** yields the best performance, indicating that the attention on image or queries has more effects on the A-ATT mechanism than the attention on objects does.

5. Conclusions and future works

In this paper, we have proposed a novel accumulated attention (A-ATT) mechanism to ground the natural language query into the image. Our model considers three types of attention, *i.e.*, query attention, image attention and objects attention. Considering that the three kinds of attention are often highly related to each other and each type of attention can be strengthened by the other two, we propose the A-ATT mechanism to circularly refine the attention for information transferring and accumulation. In this way, the noises and redundancy will decrease gradually, leading to an improved performance. Our model is able to deal with various types of queries, ranging from short phrases to long dialogues. We evaluate the effectiveness of the proposed method on four popular datasets. Extensive experiments demonstrate the state-of-the-art performance over existing methods.

There exist several open questions to explore in the future. First, it is necessary and important to improve the robustness of the model in more complex problem settings. More specifically, more round of A-ATT mechanism should show more advantages in a complex setting. Second, currently, we consider only the information from queries and images. However, it is valuable to improve the attention accumulation mechanism using additional information from other sources such as larger databases and web. With more source of information, we can design more kinds of attention modules, and build a more sufficient interaction within those information through the proposed A-ATT mechanism, which is expected to achieve an even better performance.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China (NSFC) 61502177, 61472267 and 61602185, and Recruitment Program for Young Professionals, and Shenzhen Chuangke Foundation of China CKCY2016082919273553,and Guangdong Provincial Scientific and Technological funds 2017B090901008, 2017A010101011, 2017B090910005, and Fundamental Research Funds for the Central Universities D2172500, D2172480, and Pearl River S&T Nova Program of Guangzhou 201806010081 and CCF-Tencent Open Research Fund RAGR20170105, and Jiangsu Prov. Key research and development plan BE2017663.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012. 4

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. 2

[3] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4466–4475, 2017. 1, 3, 4, 5, 6

[4] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 2

[5] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng. Stylenet: Generating attractive visual captions with styles. In *CVPR*, 2017. 2

[6] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017. 2

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[8] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015. 2

[9] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 1, 2

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 448–456. JMLR Workshop and Conference Proceedings, 2015. 4

[11] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 5

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

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

[14] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016. 2, 3, 5

[15] R. Luo and G. Shakhnarovich. Comprehension-guided referring expressions. *Conference on Computer Vision and Pattern Recognition*, pages 3125–3134, 2017. 6

[16] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9, 2015. 2

[17] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 11–20, 2016. 1, 2, 6

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. 6

[19] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016. 6

[20] 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*, pages 91–99, 2015. 4

[21] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016. 2

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 5

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017. 2

[24] P. Wang, Q. Wu, C. Shen, and A. van den Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3909–3918, 2017. 2

[25] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 203–212, 2016. 2

[26] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 2

[27] Q. Wu, P. Wang, C. Shen, I. Reid, and A. v. d. Hengel. Are you talking to me? reasoned visual dialog generation through adversarial learning. *arXiv preprint arXiv:1711.07613*, 2017. 2

[28] F. Xiao, L. Sigal, and Y. J. Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *2017*

IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 5253–5262, 2017. 1, 2

[29] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 2, 3

[30] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4651–4659, 2016. 2

[31] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. pages 69–85, 2016. 1, 2, 4, 5, 6

[32] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3521–3529, 2017. 1, 2, 4, 6

[33] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 4