SomethingFinder: Localizing undefined regions using referring expressions

Sungmin Eum*† David Han* Gordon Briggs‡
*US Army Research Lab †Booz Allen Hamilton ‡US Naval Research Lab
eum_sungmin@bah.com david.k.han.civ@mail.mil gordon.briggs@nrl.navy.mil

Abstract

Previous research on localizing a target region in an image referred to by a natural language expression has occurred within an object-centric paradigm. However, in practice, there may not be any easily named or identifiable objects near a target location. Instead, references may need to rely on basic visual attributes, such as color or geometric clues. An expression like "a red something beside a blue vertical line" could still pinpoint a target location. As such, we begin to explore the open challenge of computational object-agnostic reference by constructing a novel dataset and by devising a new set of algorithms that can identify a target region in an image when given a referring expression containing only basic conceptual features.

1. Introduction

From object classification, visual scene description, to visual question answering, there have been tremendous strides made in deep learning based computer vision algorithms over the last decade. These advances in computer vision capabilities also include the task of reference resolution: identifying a specific location or pointing out a specific entity within an image from a given referring expression [6]. Resolution of visual reference is an essential part of human communication and collaboration, and there have been various efforts to develop methods to enable automated visual reference resolution. However, despite these advances, the state-of-the-art in computational reference still lags far behind human capabilities.

One of the key limitations of present work in computational reference is that most work relies on well-defined objects as both targets for reference or anchors in their referring expressions. However, in many instances, there may not be any easily identifiable objects to reference intended targets as shown in Figure 1. Additionally, as computer vision algorithms are deployed on mobile, robotic agents, future algorithms will be faced with the need to perform reference resolution on increasingly complex scenes in real-world environments with potentially novel, unknown objects, or the absence of normal visual structure (e.g., post-disaster recovery). In such situations, people can still refer with attributes such as colors, textures, or geometric clues such as lines or corners. For instance, a phrase such as "the small yellow stuff left of that blue corner" may be sufficient for a person to pinpoint the target location despite the absence of an anchoring object.

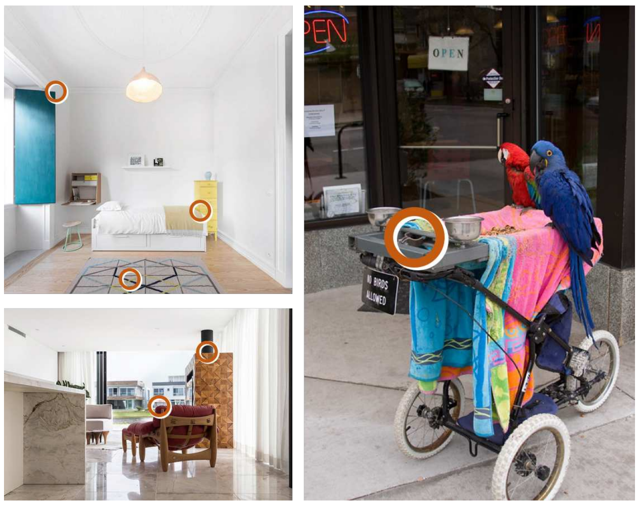
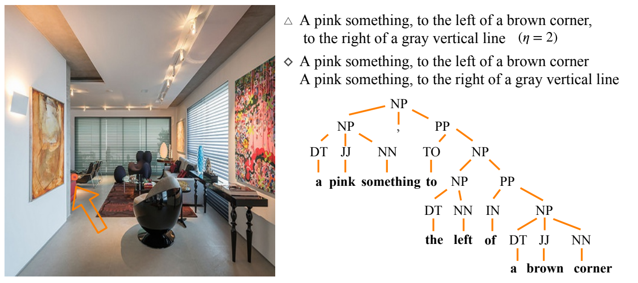


Figure 1. **Undefined or object-agnostic locations.** How would you describe to refer to these (labeled with orange circles) locations?

Thus, broadening the capability of computational systems of reference resolution requires the development of "object-agnostic" algorithms that resolve references to particular points in a visual scene that do not refer to or are anchored in well-defined objects. One difficulty in this task is the degree to which most data on human reference relies on object-centric tasks. For instance, psycholinguistic studies of reference resolution have used interactive tasks or abstract stimuli that involved references to specific objects [16]. Likewise, studies of reference generation by human subjects have also generally elicited referring expressions that refer to specific visual objects in a scene [18, 15, 7]. Some initial work has stepped beyond object-centric reference by examining how people refer to *groups* of objects [2]


△ A pink something, to the left of a brown corner, to the right of a gray vertical line ($\eta = 2$)

◇ A pink something, to the left of a brown corner
A pink something, to the right of a gray vertical line

Figure 2. **An example referring expression from Something-W.** Referred region is labeled with an orange rectangle. (An arrow is included to assist the readers.) The original expression (△) is divided into η expressions (◇) with a common phrase.

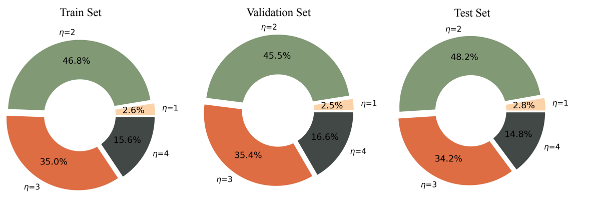


Figure 3. **Something-W Composition.** Referring expressions with complexity (η) of 2 and 3 take up approximately 80% of the dataset.

and how this can be modeled computationally [4], though this still does not address the problem of how to refer to any arbitrary region in an image.

We propose to begin to tackle the open challenge of computational object-agnostic reference in the following way. First, because there is no existing data on how people refer to points in a visual scene without recourse to objects, we present a novel dataset, called "**Something-W**," composed of images with referable conceptual features such as lines and their directions, corners, and colors. Next, we developed a novel algorithm capable of identifying candidate regions within a given image from a referring expression utilizing only the aforementioned fundamental features.

2. Dataset

There is no existing dataset designed for training on the conceptual (object-agnostic) attributes and the corresponding referring expressions. Instead of starting from scratch, we began with Wireframe [10] dataset, originally devised to train models to parse wireframe in images, as it readily provides annotations for lines. In addition to the provided line locations, we have added line directions, corners and color information which are all used to automatically generate the referring expressions for local blocks in the images.

2.1. Something-W

The total number of the referring expressions in Something-W is 25K which consists of 15K, 5K, and 5K referring expressions for train, validation, and test, respectively. Following the revision of the Google RefExp dataset [13] by Nagaraja et al. [17] we have designed the Something-W so that there is no overlap of images between different splits.

An example referring expression which refers to a local region (*block*) in an image is shown in Figure 2. Each block can be referenced based on the fundamental attributes of itself and its 4-neighbors (i.e., left, right, above and below) which are of the same size. Thus, the complexity of a referring expression can be scaled from one to four depending on how many of the neighboring regions were referenced in the expression.

Figure 3 depicts the ratio of the referring expressions with respect to their complexities (η). While the referring expressions in our train set are extracted from 180 images of the train set of the original Wireframe dataset, those in our validation and test sets are generated from another 129 images of the test set of the same dataset.

2.2. Annotations

Color. We have exploited a color palette which includes 570 English color names and their RGB values [14]. Nearest colors for each block is acquired using RGB values. To match with more realistic scenarios where people would use a smaller color vocabulary, we have post-processed the initial color names so that they would be tied with one of 21 (19 from [1], white, and black) frequently used color names.

Lines and corners. As line annotations are provided in the original Wireframe dataset, we have exploited this information to further extract the line directions and corner (junction) points. If a line falls within $\pm 15°$ range of $90°$, $0°$, and $45°$, it is categorized as *vertical*, *horizontal*, or *diagonal*, respectively. A location where two or more lines intersect is annotated as a corner point.

Referring expressions. One referring expression is automatically generated based on the previously annotated fundamental attributes (color, lines, corners) of the target block itself and its 4 neighbors. The color label of a 'self' block is used to generate the *something* phrase of a sentence (e.g., "a yellow something"), while various attribute combinations of the neighboring blocks are translated into supporting descriptions with corresponding prepositional phrases. An expression is only included into the dataset when at least one of the neighboring blocks contain reasonably salient line or corner attribute. Saliency is determined by thresholding on the previously generated corner or line maps. If one or two salient neighboring blocks exist, same number of supporting phrases are generated. However, if there exist 3 or more of them, the number of supporting phrases are randomly determined to be an integer between 2 and the number of salient neighbors. A preposition is randomly selected from a subset of a list provided by Landau et al. [11]. These

Prepositional Phrases

Left	= "to the left of", "on the left side of"
Right	= "to the right of", "on the right side of"
Above	= "above", "on top of", "over"
Below	= "below", "beneath", "under", "underneath"

Something Phrases

"something", "thing", "stuff", "region", "portion", "location", "area", "part"

Expr = a + [color] + [something] + [preposition] + [neighbor attribute] + [neighbor]

Figure 4. **Generating an expression.**

measures are designed to inject more natural flavor to the dataset. The list of prepositions and something phrases we have exploited is provided in Figure 4.

3. Our Approach

We propose a novel set of algorithms and an architecture to train the syntactic constituents of the expression with corresponding modules focusing on localizing the target with its neighboring blocks. While the backbone architecture of our model partially inherited the GroundNet [5], we developed a set of novel modules to enable visual references for the new object-agnostic paradigm.

Modules. The overall diagram of our approach is shown in Figure 5 where four major modules are depicted. We have newly devised `Visualize` and `LookAround` while `Attend` and `Locate` are inherited from the GroundNet. These modules function within a computation graph (elaborated in detail in [5]) which intakes a constituency-based parse tree (Figure 2) of a given expression.

`Visualize`. Unlike the previously introduced 'object-centric' approaches which consider a few number of objects where each object region covers up a relatively large portion in the input image, our approach involves enormously large number of regional locations (i.e., $56^2 = 3136$ blocks) which correspond to very small regions in the input images, i.e., 4×4 in our baseline setting. Because of this, instead of exploiting high level features of object regions (e.g., fc7 in [19]), we acquire low level deep CNN features which are of two dimensional in nature. We have used the output from Conv1 or Conv2 layer (both 56×56) of ResNet50 [9] pre-trained on MS COCO Dataset [12]. Each 1×1 region in the output of Conv1 ($1 \times 1 \times 64$) or Conv2 ($1 \times 1 \times 256$) corresponds to a 4×4 region in the resized (224×224) input image. The encoded features of the candidate block (ψ_c) and the neighboring blocks (ψ_n) are fed into `Locate` and `LookAround`. It is noteworthy to mention that making use of such low level features from which are known to carry crucial low level information [20] actually serve as an advantage for our approach as the fundamental attributes we consider are closely correlated with basic shapes such as corners, lines or colors.

`LookAround`. When referring expressions involve low-level attributes, regions which are distant from the target location is hardly ever mentioned. Instead, describing the regions in the close vicinity of the target is often taken into considering because there may be numerous regions which

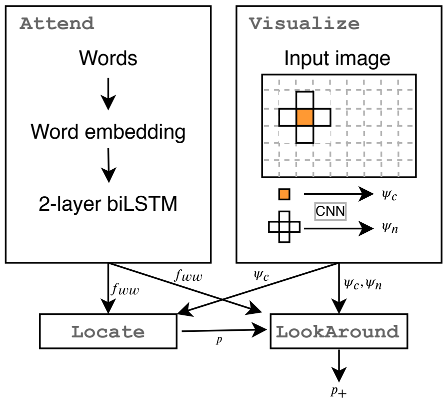


Figure 5. **Overall Diagram of our approach.** Four neural modules are presented. p_{t+} is the output score vector where each element represents the score of a box given a a list of words.

share similar low-level attributes across the scene.

Based on such motivation, we devise a module, `LookAround`, which predicts the likelihood of the blocks given a textual expression which conveys a relationship such as *on the left of* or *below*, while taking the visual features of the itself and its neighboring blocks into consideration. As can be seen in Figure 5, this module intakes the encoded text feature (f_{ww}) from `Attend`, probability vector p from `Locate`, and visual features (ψ_c, ψ_n) from `Visualize` and outputs a proabability score vector p_+. We have designed two versions of `LookAround` variant upon how the input features (visual and textual) are aggregated.

When *SepEnc* is used, the textual feature f_{ww} is encoded (W^c and W^n) with the candidate block and the neighboring blocks in a 'separate' manner as listed in Equations 1a to 1f. In this case, linear weights (W^c, W^n, W^{score}) are being learned. Symbols Θ, $\odot$, and $[\ ]$ are used to represent L2 normalization, Hadamard product, and concatenation, respectively. This process of acquiring a score s for a single box is done for all the available boxes to construct a score vector, which is then combined with p from `Locate` to provide the final probability score p_+.

$$\psi_n = [\psi_L, \psi_R, \psi_T, \psi_B] \tag{1a}$$
$$f^c_{ref} = \Theta(W^c f_{ww}) \tag{1b}$$
$$f^n_{ref} = \Theta(W^n f_{ww}) \tag{1c}$$
$$f^c_{enc} = \psi_c \odot f^c_{ref} \tag{1d}$$
$$f^n_{enc} = \psi_n \odot f^n_{ref} \tag{1e}$$
$$s = W^{score}[f^c_{enc}, f^n_{enc}] \tag{1f}$$

When *AggEnc* (Equations 2a to 2d) is deployed instead, the textual feature f_{ww} is combined with the encoded version (f_{vis}) of the concatenated (aggregated) visual feature ($[\Phi_{\psi_n}(\psi_c), \psi_n]$) at a later stage in the module. Operation $\Phi_a(b)$ is used to expand b to match the dimension of a.

Table 1. **Top-K Accuracy.** Accuracy (%) for all referring expressions in the Test set. Models are trained from scratch.

Method	Vis. feat	K=1	K=5	K=10
AggEnc	conv1	8.4	29.6	45.2
	conv2	11.0	35.9	53.6
SepEnc	conv1	8.5	31.7	49.4
	conv2	9.0	32.4	50.1
NeiEnc	conv1	7.3	26.2	41.8
	conv2	8.9	32.6	51.1

$$\psi_n = [\psi_L, \psi_R, \psi_T, \psi_B] \tag{2a}$$
$$f_{vis} = W^{agg}[\Phi_{\psi_n}(\psi_c), \psi_n] \tag{2b}$$
$$f_{agg} = \Phi_{f_{vis}}(f_{ww}) \odot f_{vis} \tag{2c}$$
$$s = W^{score} f_{agg} \tag{2d}$$

4. Experiments

Settings. For training, we have used stochastic gradient descent for 6 epochs with initial learning rate of 0.01 and learning rate decay of 0.5. The size of the hidden layer of the LSTM was set to be 500. All the weights were initialized using Xavier [8] with weight decay of 0.0005.

Evaluation Metric. As have used by previous referring expression approaches, we make use of the standard metric of accuracy. Along with the exactly correct cases (K=1), Top-K accuracy for K=5 and K=10 cases are also reported. When the ground truth location falls within the Top-K predictions made by the model, it is counted a hit.

Regular Training. Top-K accuracy for different models and settings are reported in Table 1. On top of *AggEnc* and *SepEnc*, we have also reported the performance by another variant of the LookAround module *NeiEnc* which was implemented to leave out the incorporation of ψ_c and only consider the neighboring blocks(ψ_n). The results show that conv2 features are more effective in all cases. Aggregating the candidate box and the neighboring boxes at an earlier stage (i.e., *AggEnc*) works better than other encoding methods. In this scenario, train samples are trained at a random order for each epoch disregarding the complexity (η).

Curriculum Learning. We demonstrate how our model benefits from curriculum learning [3]. Instead of learning all the available referring expressions at once, we started out with easier train samples ($\eta = 1$ and 2) for the first epoch and then introduced samples with $\eta = 3$ and $\eta = 4$ with each new epoch. Table 2 shows that not only does the overall accuracy increase in all the K cases, but the improvement is achieved at an earlier epoch demonstrating the effectiveness of the curriculum learning.

Table 2. **Curriculum Learning.** Accuracy (%) for the Test Set. Experiments were performed on *AggEnc* with conv2.

Top-K	Regular @epoch=6	Curriculum @epoch=3	Gain
K=1	11.0	11.8	+0.8
K=5	35.9	36.5	+0.6
K=10	53.6	53.9	+0.3

References

[1] Material palette. www.materialpalette.com.

[2] Dale Barr, Kees Van Deemter, and Raquel Fernández. Generation of quantified referring expressions: evidence from experimental data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 157–161, 2013.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[4] Gordon Briggs and Hillary Harner. Generating quantified referring expressions with perceptual cost pruning. In *Proceedings of the 12th International Conference on Natural Language Generation*, 2019.

[5] Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. In *AAAI*, 2018.

[6] Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. Visual referring expression recognition: What do systems actually learn? In *NAACL*, 2018.

[7] Alasdair Clarke, Daniel Francis, Micha Elsner, and Hannah Rohde. Where's wally: the influence of visual salience on referring expression generation. *Frontiers in psychology*, 4:329, 2013.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[10] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *CVPR*, 2018.

[11] Barbara Landau and Ray Jackendoff. "what" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 1993.

[12] 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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, 2014.

[13] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana-Maria Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[14] Wilson Mar and Ayush Pareek. Optimized rgb-to-colorname api. https://github.com/ayushoriginal/Optimized-RGB-To-ColorName, 2016.

[15] Margaret Mitchell, Kees van Deemter, and Ehud Reiter. Natural reference to objects in a visual domain. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 95–104. Association for Computational Linguistics, 2010.

[16] Aparna S Nadig and Julie C Sedivy. Evidence of perspective-taking constraints in children's on-line reference resolution. *Psychological Science*, 13(4):329–336, 2002.

[17] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016.

[18] Thomas Pechmann. Incremental speech production and referential overspecification. *Linguistics*, 27(1):89–110, 1989.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NEURIPS*, 2015.

[20] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, 2014.