7-23-2019

Analyzing the Visual Grounding of "Referring Relationships"

Kennedy Hahn
Portland State University

Analyzing the Visual Grounding of "Referring Relationships"

by
Kennedy Hahn

An undergraduate honors thesis submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Science
in
University Honors
and
Computer Science

Thesis Advisor
Melanie Mitchell, Ph. D.

Portland State University
2019

Analyzing the Visual Grounding of "Referring Relationships"

Kennedy Hahn
Computer Science Department
Portland State University
kehahn@pdx.edu

Abstract

There have been numerous efforts to accomplish the task of visual grounding (Deng et al., 2018, Johnson et al., 2015, Krishna et al., 2018), the act of matching regions or objects within an image with natural language queries. But with each method released, there is a growing uncertainty about the effectiveness of the machine's learning. Are computers learning what we expect, and are datasets properly testing this learning? (Cirik et al., 2018). In this thesis, I analyze the visual grounding method of "Referring Relationships" (RR) by Krishna et al. (2018). I find that RR's relationship information does not have a significant positive impact on performance as compared to a baseline model that only detects objects. In addition, I find that the Visual Relationship Detection dataset (VRD), one of the datasets used in the original paper, exhibits bias. In other words, it allows methods that do not utilize relationships to perform well, showing that the VRD dataset is not able to properly test the RR method.

1 Introduction

In the pursuit of advancing technology, it is the hope of computer scientists to have machines emulate the human brain. As such, there are many human abilities that have been implemented for computers, such as the task of performing inference on visual inputs. Just as humans can immediately interpret what they see, researchers have created methods for machines to do the same in the form of visual grounding, the act of matching regions or objects within an image with natural language queries.

Visual grounding is a challenging task. It requires that systems learn more than how to detect objects in images. The machine must be able to digest visual features of objects as well as spatial or relational models for localization, and use them to decide which object is being described in the query. Sometimes objects are small or slightly concealed in images, or numerous objects of the same category are present in one image. Queries can have different structures that systems must decipher. These issues, and many others, make visual grounding a difficult task.

There have been numerous efforts to accomplish the task of visual grounding, e.g., (Deng et al., 2018, Johnson et al., 2015, Krishna et al., 2018). But with each method released, there is a growing uncertainty about the effectiveness of the machine's learning. Are computers learning what we expect, and are datasets properly testing this learning? (Cirik et al., 2018). To answer this question, researchers are re-examining existing visual grounding methods, e.g., (Conser et al., 2019).

Likewise, this thesis revisits a particular visual grounding method called "Referring Relationships," released by (Krishna et al., 2018). I check how effectively the system learns to perform visual grounding by comparing its results with those from a simple model—one that only detects objects in an image. In addition, I use the simple model's results to check one of the datasets used by Krishna et al., to see if it allows models without relationships to perform well, and then conclude by discussing results and future work.

Figure 1: Example of RR task taken from Krishna et al. (2018). Best viewed in color.

2 Referring Relationships

The "Referring Relationships" (RR) task is to take an image and relationship query as input, and output a grounding, or set of bounding boxes for the objects of the query. This grounding in particular would produce boxes that match the query's predicate (Krishna et al., 2018). For example, given the relationship query, "person kicking ball," and an image, RR would produce bounding boxes around the person and ball objects that match the relationship query (seen above in Figure 1). RR sets itself apart from other visual grounding methods in that its main goal is to disambiguate objects in an image by "localizing the [objects] involved in the relationship" (Krishna et al., 2018). For example, if there were two "ball" objects in an image, RR would be able to differentiate between the two by looking at the relationships they were in, i.e., "ball-on-table" versus "person-kicking-ball." As seen in this example, RR defines relationships as a <Subject – Predicate – Object> (Krishna et al., 2018), where the "Subject" and "Object" in the relationship are both objects in an image.

2.1 The Method

RR receives images and their corresponding relationship queries as input, and outputs visual groundings of the queries in the image. The groundings consist of bounding boxes around the "Subject" and "Object" of each relationship query, localized by the predicate of that query. The method assumes that all natural language query inputs have the structure of <Subject—Predicate—Object> (Krishna et al., 2018).

RR trains on a dataset of images and their corresponding annotations. These annotations, created by humans, consist of all the relationship queries per image along with bounding boxes for the query objects. The training consists of 30 epochs at a 0.0001 initial learning rate. RR uses a convolutional neural network (CNN) to create a feature map for an input image (Krishna et al., 2018). In this case, a CNN is a model that can detect visual features of an image, with a feature map defined as a map of the detected features. RR also learns two "attention shift" models that are both CNN's: one learns models for the relationship from the subject to the object, and the other learns the relationship from the object to the subject. These models are called the predicate shift and inverse predicate shift models, and are learned for each predicate in the training dataset (Krishna et al., 2018).

Once the training process is done, RR is ready to be run on test images. When given an image, RR first produces initial, separate attentions for the "Subject" and "Object" (Krishna et al., 2018). Attentions represent the location of the CNN's prediction. This is done by going through each region of the image, and using an optimization model to decide whether the region

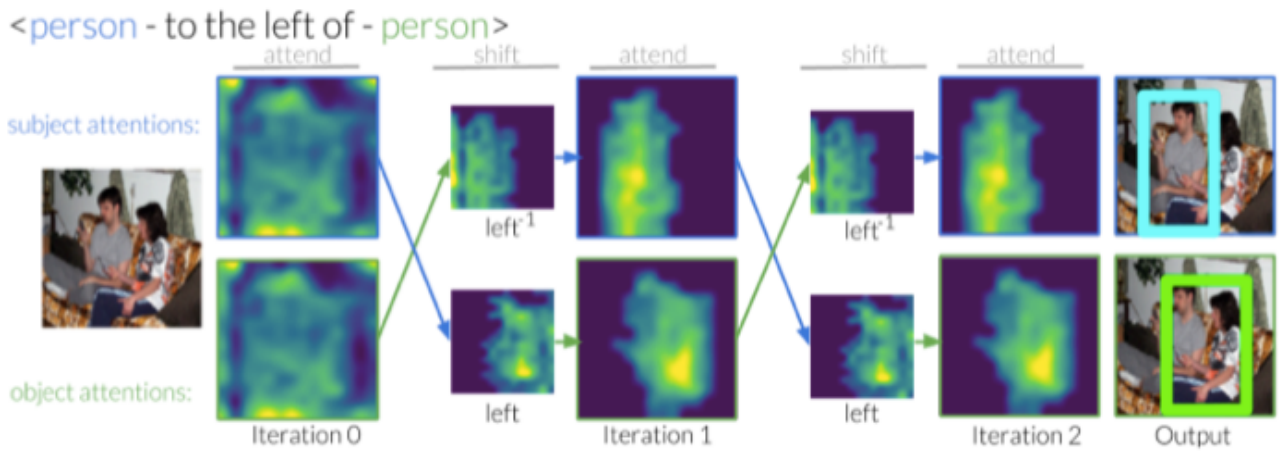

Figure 2: Example of predicate shift process taken from (Krishna et al., 2018). Best viewed in color.

depicts the "Subject" or "Object" (Krishna et al., 2018). The optimization model takes two random points in the image, X and Y, to represent the "Subject" and "Object." It then goes through the regions of the image to determine if X or Y is greater than a specified threshold. Depending on which meets that condition, the region thus depicts the "Subject" or "Object" (Krishna et al., 2018).

Once the initial attentions for "Subject" and "Object" are produced, the predicate is used to refine those attentions (Krishna et al., 2018). RR does not use the predicate as an appearance model, because the same predicate can look different in numerous, different relationships. (Krishna et al., 2018) uses the example of "person carrying phone" versus "truck carrying hay." Instead, the predicate is treated like an attention shifting model (Krishna et al., 2018). The predicate shifts the attention from the "Subject" to the "Object," and the attention from the "Object" to the "Subject," the inverse direction, is shifted by the "inverse predicate" (Krishna et al., 2018). This process is iterated multiple times for the best possible results. An example of the process is seen above in Figure 2. For the relationship, "person to the left of person," the "Subject" attentions shift left (since the predicate is "left") and the "Object" attentions shift right (the inverse predicate is "right"). In this way, the object locations become more refined until the final groundings are produced (Krishna et al., 2018).

After going through the predicate shifting models, the localized "Subject" and "Object" attentions are used to generate bounding boxes around the objects in the image, producing the final output as exemplified in Figure 1.

2.2 The Dataset

In (Krishna et al., 2018), multiple datasets were used for testing RR. This thesis focuses on one of them: the Visual Relationship Detection (VRD) dataset. VRD consists of 5,000 images, split into train and test sets (4,000 and 1,000 images respectively). It includes 100 object categories and 70 predicate categories, with a total of 38,000 relationship queries (Krishna et al., 2018). Examples of the queries included are "person behind person," or "table next to cat."

Along with the images, VRD comes with two annotation files for the train and test sets. As described in the previous section, these files contain the annotations of all relationship queries per image. For each query, the file lists the labels of the "Subject," "Object," and "Predicate," as well as ground truth bounding boxes for the "Subject" and "Object." Ground truth bounding boxes surround the correct "Subject" or "Object" in the specified relationship.

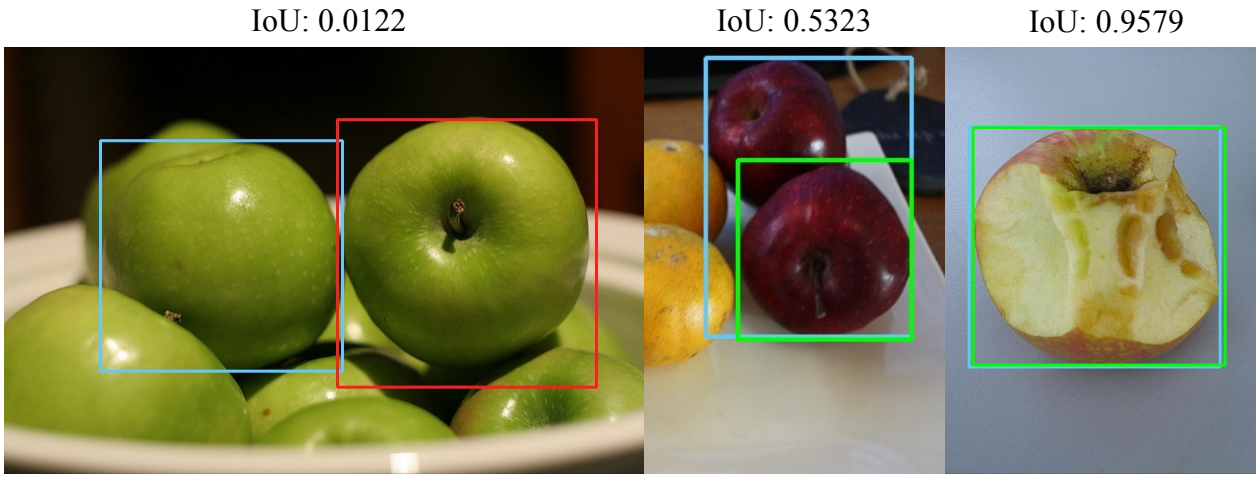

Figure 3: Visualization examples of IoU scores. Best viewed in color.

2.3 Results Stated in Paper

The original results of Krishna et al. were given as the mean intersection over union (IoU) over all "Subjects" and all "Objects" with their ground truth bounding boxes. IoU is calculated by dividing the intersection of the predicted and ground truth boxes by the union of the two. It is used as a metric to see how close a model's predicted bounding box is to the actual ground truth box. Example illustrations of IoU can be seen above in Figure 3, where the red and green boxes are predictions, and the blue boxes are the ground truth.

In RR, the authors reported IoU scores of 0.369 over all "Subjects," and 0.410 over all "Objects," after 3 iterations (Krishna et al., 2018).

3 Analyzing RR

For this thesis, I tried to replicate the results of Krishna et al., and compare them with results produced by a model that did not use relationships. I also checked if the VRD dataset contains bias. Bias is present when models that do not use relationships still perform well on the dataset. In conducting these experiments, I hoped to check how effectively RR was utilizing relationships to perform visual grounding, and how accurately the VRD dataset was testing RR.

3.1 Faster-RCNN

Faster-RCNN (Faster-Region Convolutional Neural Network) was used as the "object only" model for this thesis. Simply put, faster-RCNN is an object detector – a model that detects objects in an image. It consists of a region proposal network that produces region proposals to be used by another network to detect objects. For each predicted object, faster-RCNN provides a confidence score to represent how "confident" it is that the prediction is correct.

I obtained faster-RCNN results as follows. First, I obtained a pre-trained faster-RCNN model, faster_rcnn_resnet101_coco, from a Tensorflow GitHub repository.[1] I then fine-tuned the faster-RCNN model. In this case, fine-tuning consisted of training the outer layers of faster-RCNN to detect the 100 object categories of the VRD dataset. I then ran it on the VRD test dataset, getting the bounding boxes and their associated confidence scores of all the objects it detected per image. With those bounding box predictions and

[1] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

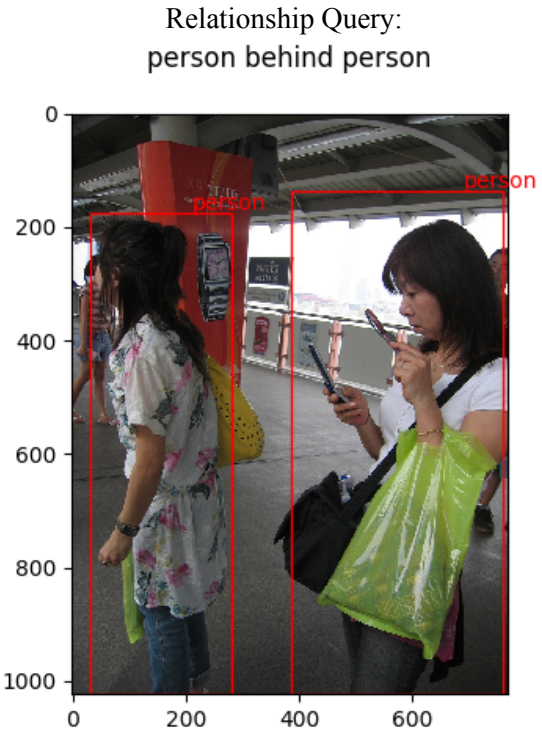

Figure 4: Example of faster-RCNN visualization. Best viewed in color.

confidence values, I was able to complete two tasks: 1) visualize each relationship per image, and 2) calculate the mean IoU over all "Subjects" and all "Objects."

To visualize each relationship per image, I chose the highest-confidence predicted box for the "Subject" and "Object" respectively (or the highest and second highest-confidence boxes if the "Subject" and "Object" were the same object class). Figure 4 shows one of the visualizations created from this process.

This was done for each relationship per image separately—in other words, the same image could appear several times with a different relationship visualized in each. In total, there were 4,710 visualizations. I then went through each visualization manually to see if the two objects detected correctly depicted each relationship query. If the boxes depicted objects in the relationship I was looking for, I labelled the image "yes" (as in correct), or "no" (meaning incorrect) for the opposite. I saw that 2,632, or ~55.88% of the visualizations looked correct, while 2,078, or ~44.12% of the visualizations looked incorrect.

To calculate the mean IoU, I again took the highest-confidence boxes for the "Subject" and "Object" to use with the ground truth "Subject" and "Object" boxes. I calculated the mean IoU for the "Subject" boxes and "Object" boxes separately, averaging the "Subject" IoU's and "Object" IoU's. The mean IoU over all "Subjects" was ~0.2697, and the mean IoU over all "Objects" was ~0.2766.

3.2 Replicating the Original Results

I downloaded the code from Krishna et al.'s public GitHub repository[2], and started training the model according to their provided instructions on GitHub. During this process, the training seemed to be extremely slow, with the time it would end unforeseeable. Despite using a GPU instead of a CPU, I could not get the training to finish quickly enough. Fortunately, I knew that another intern,

[2] https://github.com/StanfordVL/ReferringRelationships

Relationship Query:
Final Bounding Boxes: person-behind-person

Figure 5: Example of RR visualization. Best viewed in color.

Chandler Watson, had finished training the model in a previous term, and he agreed to send me his trained model.

Once I obtained the trained model, I ran it on the VRD test set, obtaining the predicted bounding boxes to create visualizations and calculate mean IoU as I did with the faster-RCNN boxes. In the process of creating visualizations, I found that the images became extremely low resolution after being rescaled to fit the bounding boxes. To remedy this, I instead rescaled the bounding boxes to fit the dimensions of the image height and width. In total, 7,632 visualizations were produced. This number was too large to manually look through by one individual, so I took a sample size of 1,000 to look for accuracy. Figure 5 shows an example of a visualization produced from this process.

In looking through the visualizations, I noticed that many images were missing either the "Subject" box or the "Object" box. These visualizations were thus labelled as incorrect, causing the number of incorrect visualizations to outnumber the correct ones. Overall, I saw that 412, or 41.2% of the visualizations correctly found the objects that matched the relationship query, while 588, or 58.8% of the visualizations looked incorrect. It was unclear how one of the bounding boxes was missing, as original author Ranjay Krishna did not experience this issue[3].

To calculate the mean IoU, I used the rescaled bounding boxes and the annotated ground truth boxes for the test dataset. For all "Subjects," I calculated a mean IoU of 0.3275. For all "Objects," the mean IoU was 0.3366.

[3] Personal Communication

Mean IoU	Subject	Object
faster-RCNN	0.2697	0.2766
RR (My replication of Krishna et al., 2018)	0.3275	0.3366
RR (Krishna et al., 2018)	0.3690	0.4100

Table 1: Mean IoU

3.3 Comparison

In terms of visualization comparison, RR had 41.2% relationships with correct groundings out of a 1,000 image sample size from 7,632, while faster-RCNN had a greater 55.88% out of 4,710. Faster-RCNN had fewer visualizations, because relationships were only visualized if the "Subject" and "Object" were both detected. However, as mentioned in the previous section, RR's lower score was due to the problem where only one bounding box appeared in the visualization (causing the visualization to be counted as incorrect). So, while faster-RCNN only included visualizations with both bounding boxes, RR allowed visualizations with just one. Because of this difference, comparing their two visualization scores would not provide useful inference in regards to the effectiveness of the RR method.

Apart from this, faster-RCNN's performance of over 50% was astonishing. As a simple object detector lacking any relationship data, faster-RCNN was able to correctly ground over half of the "Subjects" and "Objects" in each relationship. This showed that the VRD dataset exhibits bias in multiple forms. First, the dataset includes images containing only one of each query object, making the use of relationships unnecessary to correctly perform grounding. For instance, in matching the phrase "horse following person," the image would depict only one horse and one person, defeating the purpose of RR's method of using relationships to distinguish between objects of the same category. In addition, there could be multiple relationship queries that referred to the same two objects in numerous ways. For example, "keyboard on laptop," "keyboard attach to laptop," and "laptop has keyboard" were all ways to describe the same relationship between a laptop and keyboard. Since the image had only one laptop and only one keyboard, faster-RCNN could easily get 3 correct results without using relationships.

Another bias in the VRD dataset was that images could have multiple instances of the relationship query present. For example, an image could have many "tree under sky." So, any tree that faster-RCNN found would correctly match the query. Clearly, weaknesses such as these allow models that do not use relationships, like faster-RCNN, to perform well on the VRD dataset.

In terms of mean IoU, RR scored higher than faster-RCNN over both subjects and objects, but lower than the scores reported by Krishna et al. as seen above in Table 1. This could have been caused by shorter training time or differences in code compared to Krishna et al. While the code used was released by Krishna et al., and the training process made to follow their provided instructions, the released code could have had differences from the code Krishna et al. used to get their reported results. As RR's mean IoU for "Subjects" and "Objects" were higher than those of faster-RCNN, it would seem that RR does utilize relationships. However, given that the difference between the scores is only between ~0.06 to ~0.13, the relationships do not seem to be significantly helpful in successfully performing visual grounding.

4 Conclusions

In this thesis, I revisited the visual grounding method, "Referring Relationships" (Krishna et al., 2018). I found that this method does not have a significant improvement in performance compared to an object detector, with the relationship information having a small impact on the results. I also found that one of the datasets

used in the original paper, VRD, was biased and allowed an object detection model, which did not use relationship information, to perform relatively well.

This research is part of the larger movement in machine learning to make sure that computers are learning to perform tasks effectively at the conjunction of vision and language, and that datasets can properly test methods without bias (Cirik et. al., 2018b).

5 Future Work

In the future, I hope to produce a closer replication of the original results stated in "Referring Relationships" (Krishna et al., 2018). This will allow for better analysis and inference of the compared results. It would also be interesting to see how "Referring Relationships" performs on a dataset that is not biased. I plan to examine the other datasets mentioned in the original paper, CLEVR and Visual Genome, for bias as I did the VRD dataset, and continue to analyze other visual grounding methods for effectiveness.

Acknowledgments

I would first like to thank Dr. Melanie Mitchell for being my advisor, mentor, and guide this past year. It is because of the opportunity she continued to provide that I was introduced into the world of research and artificial intelligence, and could gain beneficial experience and learning. I would also like to thank Erik Conser, whose paper, "Revisiting visual grounding," was the inspiration and model for this thesis, and Chandler Watson, who provided his trained model when training mine had time constraints. I am so grateful to have been a part of such a team, and hope that our paths continue to cross in the future.

References

Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. 2018. Visual grounding via accumulated attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7746—7755.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2016. Visual relationship detection with language priors. In *European Conference on Computer Vision (ECCV)*, pages 852–869.

Erik Conser, Kennedy Hahn, Chandler Watson, and Melanie Mitchell. 2019. Revisiting visual grounding. In *Proceedings of the Workshop on Shortcomings on Vision and Language of NAACL-2019, ACL*, pages 37—46.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668—3678.

Ranjay Krishna, Ines Chami, Michael Bernstein,and Li Fei-Fei. 2018. Referring relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6867–6876.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018a. Using syntax to ground referring expressions in natural images. In *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI)*, pages 6756–6764.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018b. Visual referring expression recognition: What do systems actually learn? In *Proceedings of NAACL-HLT 2018*, pages 781-787.