The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Using Syntax to Ground Referring Expressions in Natural Images

Volkan Cirik, Taylor Berg-Kirkpatrick, Louis-Philippe Morency
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{vcirik,tberg,morency}@cs.cmu.edu

Abstract

We introduce GroundNet, a neural network for referring expression recognition – the task of localizing (or grounding) in an image the object referred to by a natural language expression. Our approach to this task is the first to rely on a syntactic analysis of the input referring expression in order to inform the structure of the computation graph. Given a parse tree for an input expression, we explicitly map the syntactic constituents and relationships present in the tree to a composed graph of neural modules that defines our architecture for performing localization. This syntax-based approach aids localization of *both* the target object and auxiliary supporting objects mentioned in the expression. As a result, GroundNet is more interpretable than previous methods: we can (1) determine which phrase of the referring expression points to which object in the image and (2) track how the localization of the target object is determined by the network. We study this property empirically by introducing a new set of annotations on the GoogleRef dataset to evaluate localization of supporting objects. Our experiments show that GroundNet achieves state-of-the-art accuracy in identifying supporting objects, while maintaining comparable performance in the localization of target objects.

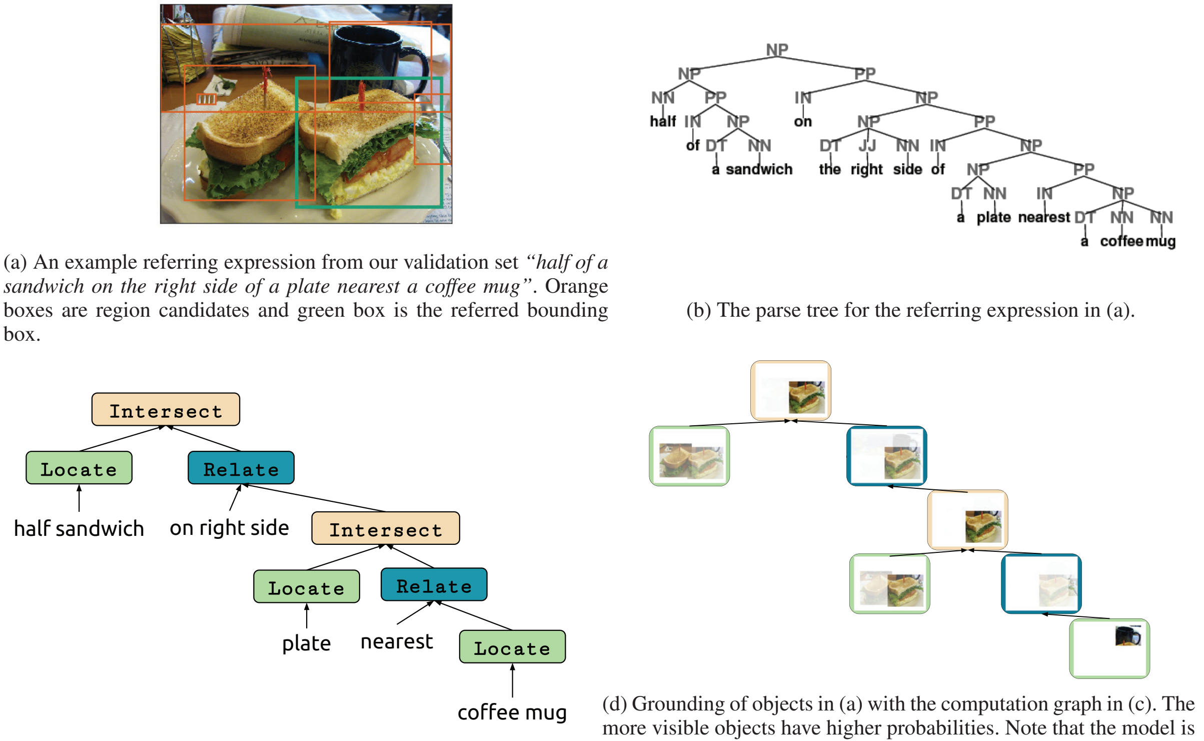
1 Introduction

Spatial referring expressions are part of our everyday social life (“Please drop me at the blue house next to the red mailbox.”) and also part of professional interactions (“Could you pass the small scalpel to the right of the forceps?”). These natural language expressions are designed to uniquely locate an object in the visual world. The process of grounding referring expressions into visual scenes involves many intermediate challenges. As a first step, we want to locate all the objects mentioned in the expression. While one of these mentions refers to the target object, the other mentions (i.e. supporting object mentions) are also important because they were included by the author of the referring expression in order to disambiguate the target. In fact, Grice (1975) argued that supporting objects will only be mentioned when they are *necessary* for disambiguation. As a second step, we want to identify the spatial relationships between these objects. Is the target to the left of the supporting object? Is it beneath it? To make effective use of an identified supporting object, we must understand how this object is related to the target. And finally, for many natural referring expressions, the process is recursive: a supporting object may itself be identified by a relationship with another supporting object. As a result, models that reason about referring expressions must respect this hierarchy, processing sub-expressions before attacking larger expressions. Modeling this compositionality is critical to designing recognition systems that behave in an interpretable way and can justify their decisions.

In this paper, we introduce GroundNet, the first dynamic neural architecture for referring expression recognition that takes full advantage of syntactic compositionality. Past approaches, such as the Compositional Modular Networks (CMN) model (Hu et al. 2017), have relied on limited syntactic information in processing referring expressions – for example, CMN tracks a single supporting object – but have not modeled linguistic recursion and therefore is incapable of tracking multiple supporting objects. As shown in Figure 1, our GroundNet framework relies on a syntactic parse of the input referring expression to dynamically create a computation graph that reflects the recursive hierarchy of the input expression. As a result, our approach tracks intermediate localization decisions of all supporting objects. Following the approach of (Andreas et al. 2016b; 2016a), this computation graph is translated into a neural architecture that keeps interpretable information at each step of the way, as can be seen in Figure 1d.

We additionally present a new set of annotations that specify the correct locations of supporting objects in a portion of the standard benchmark dataset, GoogleRef (Mao et al. 2016) to evaluate the interpretability of models for referring expression recognition. Using these additional annotations, our empirical evaluations demonstrate that GoundNet substantially outperforms the state-of-the-art at intermediate predictions of the supporting objects, yet maintains comparable accuracy at target object localization. These results demonstrate that syntactic compositionality can be successfully used to improve interpretability in neural models of language and vision. Our annotations for supporting objects and implementations are available for public use[1].

[1] https://github.com/volkancirik/groundnet



(a) An example referring expression from our validation set *"half of a sandwich on the right side of a plate nearest a coffee mug"*. Orange boxes are region candidates and green box is the referred bounding box.

(b) The parse tree for the referring expression in (a).

(c) Computation graph for the parse tree in (b).

(d) Grounding of objects in (a) with the computation graph in (c). The more visible objects have higher probabilities. Note that the model is able to ground supporting objects like the coffee mug.

Figure 1: An Overview of GroundNet. A referring expression (a) is first parsed (b). Then, the computation graph of neural modules is generated using the parse tree (c). Each node localizes objects present in the image (d).

2 GroundNet

In this section, we explain the motivation of GroundNet, how we generate the computation graph for GroundNet, and finally, detail the neural modules that we use for computing the localization the referring expressions.

Motivation

A referring expression disambiguates a target object using the object's discriminative features such as color, size, texture etc., and their relative position to other *supporting* objects. Figure 1a shows a canonical example from our task w one half of a sandwich is referred by "half of a sandwich on the right side of a plate nearest a coffee mug". Here the sandwich is disambiguated using relative clauses (e.g. "the right side of" , "nearest") and the *supporting* objects (e.g "plate", "coffee mug"). We observe that there is a correspondence between the linguistic compositional structure (i.e. the parse tree) of the referring expression and the process of resolving a referring expression. In Figure 1b, we see that the target object and supporting objects have a noun phrase (NP) on the parse tree of the referring expression. Also, the relative positioning of objects in the image (e.g. being on the right, or near) correspond to prepositional phrases (PP) on the tree. We design GroundNet based on this observation to localize the target object by modeling the compositional nature of the language. The compositionality principle states that the meaning of a constituent is a function of (i) its building blocks and (ii) the recursive rules to combine them. In our case, the building blocks for the GroundNet is grounding of objects i.e. the probability of how likely an object is for word phrases. The combining rules are defined by the parse tree describing what these objects are and how they are related to each other.

GroundNet models the processing of a referring expression in a computation graph (see Figure 1c) based on the parse tree of the referring expression (see Figure 1b). Nodes of the computation graph have 3 different types aiming to capture the necessary computations for localizing the target object. `Locate` nodes ground a noun phrase ("half sandwich", "plate","coffee mug"), i.e. pointing how likely that a given noun phrase refers to an object present in the image. For example, in Figure 1d, `Locate` node of the phrase "half sandwich" outputs higher probabilities for both halves of sandwiches compared to other objects. Prepositional phrases ("on right side","nearest") correspond to `Relate` nodes in the computation graph. `Relate` nodes calculate how likely objects are related to the grounding of objects with given prepositional phrase. For instance, in Figure 1c, the `Relate` node of "nearest" computes how likely the objects are related to the grounding of "coffee mug" with the relation "nearest". We convert the phrases coming from branches in the parse tree to `Intersect` nodes. It simply intersects two sets of groundings so that objects that have high likelihood in both branches will have high probabilities for the output (see the root node in Figure 1d). Since each node of this computation graph outputs a grounding for its subgraph, GroundNet is interpretable as a whole. At each node, we can visualize how model's multiple predictions for objects propagates through the computation graph.

In following sections, we detail how we generate the computation graph and the neural modules used in GroundNet.

Generating a Computation Graph

GroundNet processes the referring expression with a computation graph (Figure 1c) based on to the parse tree (Figure 1b) of the referring expression. First, we parse the referring expression with Stanford Parser (Manning et al. 2014). Then, we generate the computation graph (see Figure 1b, 1c for an example) for a parse tree with a recursive algorithm (see Algorithm 1).

Algorithm 1: Generate Computation Graph

```
 1: procedure GenerateComputationGraph(tree)
 2:     left_NP = FindNP(tree.left)
 3:     right_NP = FindNP(tree.right)
 4:     if left_NP == ”” then
 5:         return (Locate tree.text)
 6:     end if
 7:     Relate = FindPP(tree, [left_NP, right_NP])
 8:
 9:     left_cg = GenerateComputationGraph(left_NP)
10:     right_cg = GenerateComputationGraph(right_NP)
11:     return (Intersect (left_cg) (Relate right_cg))
12: end procedure
```

Above, the function *FindNP* finds the noun-phrase with the largest word span of given root node for left and right branches (line 2, 3). If the tree does not have an NP subtree, it returns a `Locate` node (line 5).

FindPP extracts the words between noun-phrases to model the relationship between them and returns a `Relate` node (line 7). For both left and right branches of the parse tree, the same algorithm is recursively called (lines 9, 10). Finally, the sub-computation graphs of left and right branches are merged (line 11) into an `Intersect` node.

Each node in the computation graph is decorated with the phrase T using the text span, i.e. constituents, of the corresponding parse tree node. We filter out the function words such as determiners ‘a‘ and ‘the‘. For instance, the `Locate` on the left in Figure 1c has the span of words “half sandwich” from the corresponding noun phrase “the half of a sandwich” in Figure 1b.

In the following section, we explain the set of neural modules that we design for performing the localization of the referring expression on a composed computation graph.

Neural Modules

We operationalize the computational graph for a referring expression into an end-to-end neural architecture by designing neural modules that represent each node of our graph. First, let us introduce the notation for referring expression task. For each referring expression, (I, R, X) are inputs where I is an image, R is the set of bounding boxes r_i of objects present in the image I, and X is a referring expression disambiguating a target object in bounding box r^*. Our aim is to predict r^* processing the referring expression in a computational graph with neural modules. In addition to (I, R, X), neural modules use the output of other neural modules and the text span T of the computation node.

We detail parameterization of neural modules in following subsections and visualize them in Figure 2 for clarity.

`Attend` This module induces a text representation for `Locate` and `Relate` nodes. It takes the words $\{w_i\}_{i=1}^{|T|}$ and embeds them to a word vector $\{e_i\}_{i=1}^{|T|}$. A 2-layer bidirectional LSTM network (Schuster and Paliwal 1997) processes embedded words. Both forward and backward layer representations are concatenated for both layers into a single hidden representation for each word as follows:

$$h_i = [h_i^{(1,fw)} h_i^{(1,bw)} h_i^{(2,fw)} h_i^{(2,bw)}] \tag{1}$$

The attention weights are computed with a linear projection using W^a:

$$a_i = \frac{exp(W^a h_i)}{\sum_{i=1}^{|T|} exp(W^a h_i)} \tag{2}$$

The output of `Attend` is the weighted average of word vectors e_i where the weights are attentions a_i.

$$f_a(T; \Theta_a) = \sum_{i=1}^{|T|} a_i e_i \tag{3}$$

The learned parameters Θ_a of this module are the parameters of 2-layer bidirectional LSTM and scoring matrix W^a.

`Locate` This module predicts which object is referred to for a text span, i.e. noun phrase, in the referring expression. It computes the probability distribution over bounding boxes using the output of `Attend` and feature representations of bounding boxes. For instance in Figure 1c, `Locate` node with input “half sandwich” localizes objects by scoring each bounding box. `Locate` node does so by scoring how well the text span “half sandwich” matches the content of each bounding box.

To represent a bounding box r, we use spatial and visual features. First, visual features r_{vis} for the bounding box are extracted using a convolutional neural network (Ren et al. 2015). Second, spatial features represent position and size of the bounding box. We have 5-dimensional vectors for spatial features $r_{spat} = [\frac{x_{min}}{W_I}, \frac{y_{min}}{H_I}, \frac{x_{max}}{W_I}, \frac{y_{max}}{H_I}, \frac{S_r}{S_I}]$ where S_r is the size and $[x_{min}, y_{min}, x_{max}, y_{max}]$ are coordinates for bounding box r and S_I, W_I, H_I are area, width, and the height of the input image I. These two representations are concatenated as $r_{vis,spat} = [r_{vis} r_{spat}]$ for a bounding box r.

We follow the previous work (Hu et al. 2017) for

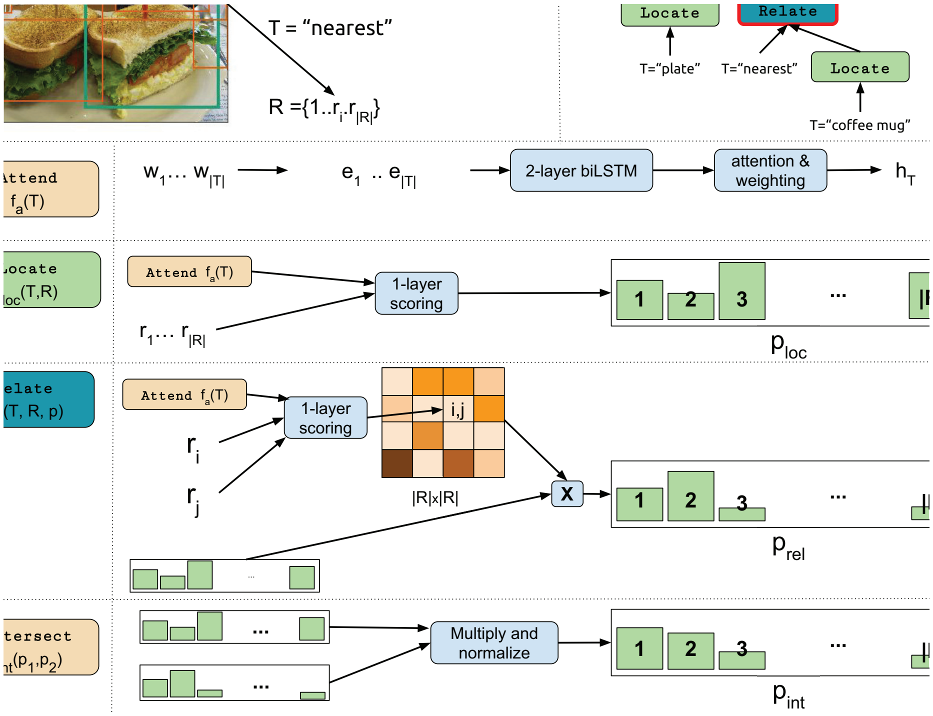


Figure 2: Illustrations of GroundNet's neural modules. Upper left shows an example referring expression and the input for `Relate` node (upper right, highlighted in red) of a small section of a computation graph. Modules take inputs from module's text span T, the set of bounding boxes R, and output probabilities of other nodes p_i. Best seen in color.

parametrization of `Locate`.

$$\hat{r}_{vis,spat} = W^{loc}_{vis,spat} r_{vis,spat} \tag{4}$$

$$z_{loc} = \hat{r}_{vis,spat} \odot f_a(T) \tag{5}$$

$$\hat{z}_{loc} = z_{loc} / \mid\mid z_{loc} \mid\mid_2 \tag{6}$$

$$s_{loc} = W^{loc}_{score} \hat{z}_{loc} \tag{7}$$

$$p_{loc} = softmax(s_{loc}) \tag{8}$$

$$f_{loc}(T, R; \Theta_{loc}) = p_{loc} \tag{9}$$

First, $r_{vis,spat}$ is projected to the same dimension as the text representation coming from the `Attend` (Eq 4). Text and box representations are element-wise multiplied to get z_{loc} for a joint representation of the text and bounding box. We normalize with L2-norm into $\hat{z}_{loc}$ (Eq 5, 6). Localization score s_{loc} is calculated with a linear projection of the joint representation (Eq 7). Localization scores are fed to softmax to form a probability distribution p_{loc} over boxes. The learned parameters Θ_{loc} of this module are the matrices $W^{loc}_{vis,spat}$ and W^{loc}_{score}.

`Relate` predicts how likely an object *relates* to the other objects with some relation described by the node's text span. For instance, the relation "nearest" in Figure 1d holds for half-sandwich pairs, and a half-sandwich and coffee mug pair. Since the incoming `Locate` node to `Relate` outputs a high probability for the coffee mug, only objects near to coffee mug have a high probability. GroundNet does so by first computing a relationship score matrix for boxes and multiplying the scoring matrix with the grounding input. We do not define a set of relationships for `Relate`, instead, model learns how objects relate to each other using module's text representation. Specifically, this module computes a relationship score matrix S_{rel} of size $R \times R$ consisting of scores for box i and j as follows:

$$\hat{r}_{i,j} = W^{rel}_{spat} r_{i,j} \tag{10}$$

$$z_{rel} = \hat{r}_{i,j} \odot f_a(T) \tag{11}$$

$$\hat{z}_{loc} = z_{rel} / \mid\mid z_{rel} \mid\mid_2 \tag{12}$$

$$S_{rel}[i,j] = W^{rel}_{score} \hat{z}_{rel} \tag{13}$$

$$p_{rel} = S_{rel} p \tag{14}$$

$$f_{rel}(T, R, p; \Theta_{rel}) = p_{rel} \tag{15}$$

Above, spatial representations of boxes are concatenated as $r_{i,j} = [r_{i,spat}, r_{j,spat}]$ and projected into the same dimension as text representation (Eq 10). Similar to `Locate`, text and box representations are fused with element-wise multiplication and L2-normalization (Eq 11, 12), then box pair is scored linearly (Eq 13).

Finally, the probability distribution p_{rel} over bounding boxes is calculated as $p_{rel} = S_{rel} p_{loc}$. The learned parameters Θ_{rel} of this module are the matrices W^{rel}_{spat} and W^{rel}_{score}.

`Intersect` This module combines groundings coming from two branches of the computation graph by simply multiplying object probabilities and normalizing it to form a probability distribution. In the following section, we explain our experimental setup.

3 Experiments

Now, we detail our experimental setup. In our experiments, we are interested in following research questions:

- **(RQ1)** How successful models are incorporating the syntax and how important the dynamic and modular computation in exploiting the syntactic information?
- **(RQ2)** What are the accuracies of models for supporting objects and how these accuracies change depending on the syntactic information?

Now, we explain datasets used for our experiments.

Referring Expression Dataset. We use the standard Google-Ref (Mao et al. 2016) benchmark for our experiments. Google-Ref is a dataset consisting of around 26K images with 104K annotations. We use ”Ground-Truth” evaluation setting where the ground truth bounding box annotations from MSCOCO (Lin et al. 2014) are used.

Supporting Objects Dataset. We also investigate the performances of models in terms of interpretability. We measure the interpretability of a model by its accuracy on both target and supporting objects. To this end, we present a new set of annotations on Google-Ref dataset. First, we run a pilot study on MTurk where all bounding boxes and the referring expression present to annotators[2]. Our in-house annotator has an agreement of 0.75 - a standard metric in word alignment literature (Graca et al. 2008; Ozdowska 2008) with three turkers on a small validation set of 50 instances. Overall, our annotator labeled 2400 instances – but only 1023 had at least one supporting object bounding box.

Number of Supporting Objects	0	1	2	3	4
Number of Instances	1377	891	118	11	3

Table 1: Statistics for the number of supporting objects for annotated 2400 instances.

We remind that the training data does not have any annotations for supporting objects. Models should be able to predict supporting objects using only target object supervision and text input. We should emphasize that our work is the first to report quantitative results on supporting object for the referring expression task and we release our annotation for future studies. Next, we provide details of our implementation.

Implementation Details. We trained GroundNet with backpropagation. We used stochastic gradient descent for 6 epochs with and initial learning rate of 0.01 and multiplied by 0.4 after each epoch. Word embeddings were initialized with GloVe (Pennington, Socher, and Manning 2014) and finetuned during training. We extracted features for bounding boxes using fc7 layer output of Faster-RCNN VGG-16 network (Ren et al. 2015) pre-trained on MSCOCO dataset (Lin et al. 2014). Hidden layer size of LSTM networks was searched over the range of {64,128,...,1024} and picked based on best validation split which is 2,5% of training data separated from training split. Following the previous work (Hu et al. 2017), we used official validation split as the test. We initialized all parameters of the model with Xavier initialization (Glorot and Bengio 2010) and used weight decay rate of 0.0005 as regularization. Next, we explain models used in our experiments.

Baseline Models. We compare GroundNet to the recent models from the literature. **RecursiveNN** Socher et al. (2014) use the recursive structure of syntactic parses of sentences to retrieve images described by the input sentence. The text representation of a referring expression is recursively calculated following the parse tree of the referring expression. The text representation at root node is jointly scored with bounding box representations and the highest scoring box is predicted. **LSTM + CNN - MMI** Mao et al. (2016) use LSTMs for processing the referring expression and CNNs for extracting features for bounding boxes and the whole image. Model is trained with Maximum Mutual Information training. **LSTM + CNN - MMI+visdif** Yu et al. (2016) introduce contextual features for a bounding box by calculating differences between visual features for object pairs. **LSTM + CNN - MIL**[3] Nagaraja, Morariu, and Davis (2016) score object-supporting object pairs. The pair with the highest score is predicted. They use Multi Instance Learning for training the model. **CMN**[4] Hu et al. (2017) introduce a neural module network with a tuple of object-relationship-subject nodes. The text representation of tuples are calculated with an attention mechanism (Bahdanau, Cho, and Bengio 2014) over the referring expression. We also report results for **CMN - syntax guided** when a parse tree is used for extracting the object-relationship-subject tuples.

GroundNet with varying level of syntax. We investigate the effect of the syntax varying the level of use of the syntactic structure for GroundNet. **GroundNet** is the original model presented in the previous section where each node in computation graph uses the node’s text span for Attend. For **GroundNet-syntax-guided Locate** model, Locate nodes use the node’s text span as an input to the Attend module. Whereas for Relate nodes can use all referring expression for inducing the text representation. For **GroundNet-free-form** model, Both Locate and Relate nodes use all of the referring expression as the input to Attend. Next, we explain our evaluation metrics used in our experiments.

Evaluation. To evaluate models for referring expression task we use the standard metric of accuracy. For evaluation of supporting objects, when there are multiple supporting objects, we consider a supporting object prediction as

[2] We did not provide the parse trees to not bias the annotators.

[3] Originally the authors use a new test split, whereas, we report results for the standard split of the dataset for this model.

[4] We report results for our reimplementation of this model where we did hyperparameter search the same as our model.

Model	Syntax	Dynamic Computation	Modularity	Relationships	Supporting(%)	Accuracy(%)
LSTM+CNN - MMI						60.7
LSTM+CNN - MMI+visdif				✓		64.0
LSTM+CNN - MIL				✓	15.0	67.3
CMN			✓	✓	11.1	69.7
Recursive NN	✓	✓				51.5
CMN-syntax guided	✓		✓	✓		53.5
GroundNet	✓	✓	✓	✓	60.6	65.7
GroundNet-syntax-guided `Locate`	✓	✓	✓	✓	60.0	66.7
GroundNet-free-form	✓	✓	✓	✓	10.6	68.9

Table 2: The accuracy of models with the support of syntax, dynamic computation, modularity, relationship modeling, and supporting object predictions. Our model is the first syntax-based model with successful results and achieves the best results in supporting object localization.

accurate only if at least one supporting object is correctly classified. To evaluate approaches modeling the supporting objects we use following methods. For LSTN+CNN-MIL, we use the context object of the maximum scoring target-context object pair as the supporting object. For CMN, we use the object with the maximum object score of a subject-relation-object tuple as the prediction for the supporting object. For GroundNet, we use the object with maximum probability as a prediction for intermediate nodes in the computation graph. In the following section, we discuss results of our experiments.

4 Results

We presented overall results in Table 2 for the compared models. We now discuss columns of the Table 2.

(RQ1) Syntax, Dynamic Computation, and Modularity. GroundNet variations achieve the best results among syntax-based models. "Recursive NN" homogeneously processes the referring expression throughout the parse tree structure. On the other hand, GroundNet modularly parameterizes multi-modal processing of localization and relationships. "CMN - syntax guided" has a fixed computation graph of a subject-relation-object tuple, whereas, GroundNet has a dynamic computation graph for each instance, thus, a varying number of computation nodes are induced. When compared to other syntax-based approaches, GroundNet results show that a dynamic *and* modular architecture is essential to achieve competitive results with a syntax-based approach.

(RQ2) Syntax for Supporting Objects. Our model achieves the highest accuracy on localizing the supporting objects when its modules are guided by syntax. "LSTM+CNN-MIL" and CMN does not exploit the syntax of the referring expression and poorly performs in localizing supporting objects. When we relax the syntactic guidance of GroundNet by letting all modules to attend to all of the referring expression, "GroundNet-free-form" also performs poorly on localizing supporting objects. These results suggest that leveraging syntax is essential in localizing supporting objects and there might be a tradeoff between being interpretable and being accurate for models. We qualitatively show a couple of instances from test set GroundNet and CMN in Figure 3. As an example, for the first instance, both GroundNet and CMN successfully predict the target object. GroundNet is able to localize both supporting objects (i.e. the girl and the disc) mentioned in the referring expression, whereas, CMN fails to localize the supporting objects. Next, we review the previous work related to GroundNet.

5 Related Work

Referring expressiong recognition is a well-studied problem in human-robot interaction (Chai, Hong, and Zhou 2004; Zender, Kruijff, and Kruijff-Korbayová 2009; Tellex et al. 2011; Lemaignan et al. 2011; Fang, Liu, and Chai 2012; Williams et al. 2016). Here, we focus on more closely related studies where visual context is a rich set of real-world images or language with rich vocabulary is modeled with compositionality.

Grounding Referential Expressions. The most of the recent work (Mao et al. 2016; Hu et al. 2016; Rohrbach et al. 2016; Fukui et al. 2016; Yu et al. 2016; Nagaraja, Morariu, and Davis 2016) addresses grounding referential expression task with a fixed computation graph. In earlier studies (Mao et al. 2016; Hu et al. 2016; Rohrbach et al. 2016; Fukui et al. 2016), the bounding boxes are scored based on their CNN and spatial features along with features for the whole image. Since each box is scored in isolation, these methods ignore the object relationships. More recent studies (Yu et al. 2016; Nagaraja, Morariu, and Davis 2016; Hu et al. 2017) show that modeling relationship between objects improves the accuracy of models. GroundNet has a dynamic computation graph and models the relationship between objects.

Modular Neural Architectures. Neural Module Networks (NMN) (Andreas et al. 2016b; 2016a) is a general framework for modeling compositionality of language using neural modules. A computation graph with neural modules as nodes is generated based on a parse tree of the input text.

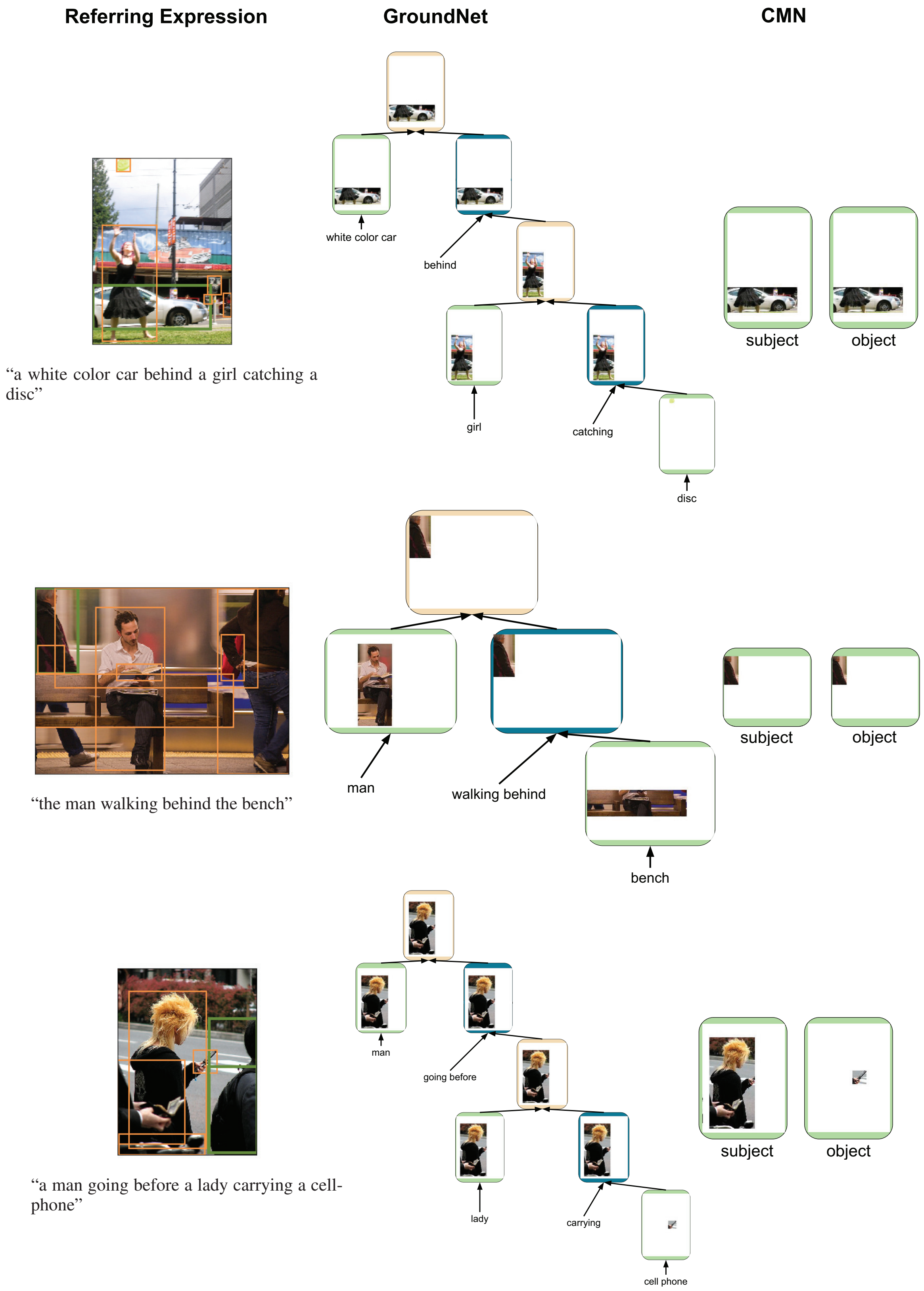


Figure 3: Qualitative Results for GroundNet. Bounding boxes and referring expressions to target object (in green boxes) on the left. GroundNet predictions in the middle and CMN predictions are on the right. GroundNet localizes not only the target object but also supporting objects (e.g. disc and girl in the first row, bench in the second).Best seen in color.

GroundNet shares the principles of this framework. We design GroundNet for referring expression task restricting each node grounded in the input image which keeps network interpretable throughout the computation.

Compositional Modular Networks (CMN) (Hu et al. 2017) is also an instant of NMN aiming to remove language parser from the generation of computation graph by inducing text representations to localization and relationship modules using an attention mechanism. Their computation graph is fixed to the subject-relation-subject tuple but the input is dynamically constructed for modules. Our model, on the other hand, can handle multiple relationships mentioned in referring expressions (see the first row of Figure 3). We should note that CMN is a special case of GroundNet where the syntax is fixed to a triplet of $Locate_{subject}, Relate, Locate_{obj}$ and each node composes a text representation with whole referring expression.

Syntax with Vision. Similar to our work, Gorniak and Roy (2004) study a syntax-based approach for grounding referring expressions. However, they use a synthetic visual scene of identical shapes with varying colors and a synthetic grammar for language. Golland, Liang, and Klein (2010) introduce a game-theoretic model successfully leverages syntax for grounding reference expressions for synthetic scenes. (Matuszek* et al. 2012) presents a semantic parsing model with Combinatory Category Grammar for referring expression recognition that jointly learns grounding of objects and their attributes. The model is able to induce latent logical forms when bootstrapped with a supervised training stage. Berzak et al. (2015) use visual context to address linguistic ambiguities. Similarly, Christie et al. (2016) use the visual context for solving prepositional phrase attachment resolution (PPAR) for sentences describing a scene. Unlike our model, their model relies on multiple parse trees and multiple segmentations of an image coming from a black-box image segmenter. Our model can also be extended to address PPAR setting where we only need to ground-truth object annotations for roots of multiple parse trees for the input sentences. Wang et al. (2016) introduce a model localizing phrases in sentences that describe an image. However, their model relies on the annotation of phrase-object pairs. GroundNet only uses target object annotations and there is no supervision for supporting objects. Xiao, Sigal, and Lee (2017) aim to address localization of phrases on region masks. Similar to our approach, they do not rely on ground-truth masks during training. However, unlike GroundNet, their model does not model relationship between objects.

6 Conclusion

In this work, we present GroundNet, a compositional neural module network designed for the task of grounding referring expressions. We also introduce a novel auxiliary task and an annotation for localizing the supporting objects.

Our experiments on a standard benchmark show that GroundNet is the first model that successfully incorporates syntactic information for the referring expression task. This syntactic information helps GroundNet achieve state-of-the-art results in localizing supporting objects. Our results show that recent models are unsuccessful at localizing supporting objects. This suggests that current solutions to referring expression task come with an interpretability-accuracy tradeoff. Our approach substantially improves supporting object localization, while maintaining high accuracy, thus representing a new and more desirable point along the trade-off trajectory.

We believe future work might extend our work with following insights. First, while generating the computation graph GroundNet, we drop the determiners. However, the indefiniteness of a noun could be helpful in localizing an object. Second, GroundNet processes the computation graph in a bottom-up fashion. An approach combining the sequential processing of the referring expression with the bottom-up structural processing of GroundNet could model expectation-driven effects of language which may result in more accurate grounding throughout the computation graph.

Acknowledgments

This project was partially supported by Oculus and Yahoo InMind research grants. The authors would like to thank anonymous reviewers and the members of MultiComp Lab at CMU for their valuable feedback.

References

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016a. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1545–1554. San Diego, California: Association for Computational Linguistics.

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 39–48.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Berzak, Y.; Barbu, A.; Harari, D.; Katz, B.; and Ullman, S. 2015. Do you see what i mean? visual resolution of linguistic ambiguities. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1477–1487. Association for Computational Linguistics.

Chai, J. Y.; Hong, P.; and Zhou, M. X. 2004. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, 70–77. ACM.

Christie, G.; Laddha, A.; Agrawal, A.; Antol, S.; Goyal, Y.; Kochersberger, K.; and Batra, D. 2016. Resolving language and vision ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1493–1503. Association for Computational Linguistics.

Fang, R.; Liu, C.; and Chai, J. Y. 2012. Integrating word acquisition and referential grounding towards physical world interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, 109–116. ACM.

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 457–468. Austin, Texas: Association for Computational Linguistics.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, 249–256.

Golland, D.; Liang, P.; and Klein, D. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 410–419. Association for Computational Linguistics.

Gorniak, P., and Roy, D. 2004. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research* 21:429–470.

Graca, J.; Pardal, J. P.; Coheur, L.; and Caseiro, D. 2008. Building a golden collection of parallel multi-language word alignment. In *LREC*.

Grice, H. P. 1975. Logic and conversation. *1975* 41–58.

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

Hu, R.; Rohrbach, M.; Andreas, J.; Darrell, T.; and Saenko, K. 2017. Modeling relationships in referential expressions with compositional modular networks.

Lemaignan, S.; Ros, R.; Alami, R.; and Beetz, M. 2011. What are you talking about? grounding dialogue in a perspective-aware robotic architecture. In *RO-MAN, 2011 IEEE*, 107–112. IEEE.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 740–755. Springer.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, 55–60.

Mao, J.; Huang, J.; Toshev, A.; Camburu, O.; Yuille, A. L.; and Murphy, K. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 11–20.

Matuszek*, C.; FitzGerald*, N.; Zettlemoyer, L.; Bo, L.; and Fox, D. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. In *Proc. of the 2012 International Conference on Machine Learning*.

Nagaraja, V.; Morariu, V.; and Davis, L. 2016. Modeling context between objects for referring expression understanding. In *ECCV*.

Ozdowska, S. 2008. Cross-corpus evaluation of word alignment. In *LREC*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 1532–1543.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

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

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Socher, R.; Karpathy, A.; Le, Q. V.; Manning, C. D.; and Ng, A. Y. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics* 2:207–218.

Tellex, S.; Kollar, T.; Dickerson, S.; Walter, M. R.; Banerjee, A. G.; Teller, S.; and Roy, N. 2011. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine* 32(4):64–76.

Wang, M.; Azab, M.; Kojima, N.; Mihalcea, R.; and Deng, J. 2016. Structured matching for phrase localization. In *European Conference on Computer Vision*, 696–711. Springer.

Williams, T.; Acharya, S.; Schreitter, S.; and Scheutz, M. 2016. Situated open world reference resolution for human-robot dialogue. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, 311–318. IEEE Press.

Xiao, F.; Sigal, L.; and Lee, Y. J. 2017. Weakly-supervised visual grounding of phrases with linguistic structures.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*, 69–85. Springer.

Zender, H.; Kruijff, G.-J. M.; and Kruijff-Korbayová, I. 2009. Situated resolution and generation of spatial referring expressions for robotic assistants. In *IJCAI*, 1604–1609.