PhraseCut: Language-based Image Segmentation in the Wild

Chenyun Wu[1] Zhe Lin[2] Scott Cohen[2] Trung Bui[2] Subhransu Maji[1]
[1]University of Massachusetts Amherst [2]Adobe Research
{chenyun,smaji}@cs.umass.edu, {zlin,scohen,bui}@adobe.com

Abstract

We consider the problem of segmenting image regions given a natural language phrase, and study it on a novel dataset of 77,262 images and 345,486 phrase-region pairs. Our dataset is collected on top of the Visual Genome dataset and uses the existing annotations to generate a challenging set of referring phrases for which the corresponding regions are manually annotated. Phrases in our dataset correspond to multiple regions and describe a large number of object and stuff categories as well as their attributes such as color, shape, parts, and relationships with other entities in the image. Our experiments show that the scale and diversity of concepts in our dataset poses significant challenges to the existing state-of-the-art. We systematically handle the long-tail nature of these concepts and present a modular approach to combine category, attribute, and relationship cues that outperforms existing approaches.

1. Introduction

Modeling the interplay of language and vision is important for tasks such as visual question answering, automatic image editing, human-robot interaction, and more broadly towards the goal of general Artificial Intelligence. Existing efforts on grounding language descriptions to images have achieved promising results on datasets such as *Flickr30Entities* [30] and *Google Referring Expressions* [26]. These datasets, however, lack the scale and diversity of concepts that appear in real-world applications.

To bridge this gap we present the VGPhraseCut dataset and an associated task of grounding natural language phrases to image regions called *PhraseCut* (Figure 1 and 2). Our dataset leverages the annotations in the *Visual Genome (VG)* dataset [18] to generate a large set of referring phrases for each image. For each phrase, we annotate the regions and instance-level bounding boxes that correspond to the phrase. Our dataset contains 77,262 images and 345,486 phrase-region pairs, with some examples shown in Figure 2. VGPhraseCut contains a significantly longer tail of concepts and has a unified treatment of stuff and object categories, unlike prior datasets. The phrases are structured into words that describe categories, attributes, and relationships, providing a systematic way of understanding the performance on individual cues as well as their combinations.

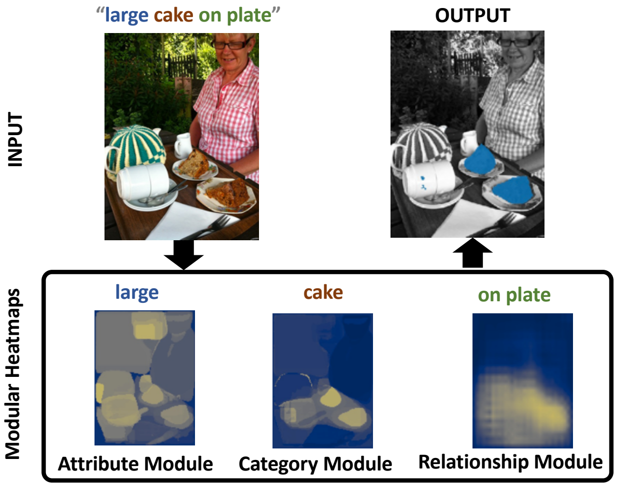

Figure 1. **Our task and approach.** PhraseCut is the task of segmenting image regions given a natural language phrase. Each phrase is templated into words corresponding to *categories*, *attributes*, and *relationships*. Our approach combines these cues in a modular manner to estimate the final output.

The *PhraseCut* task is to segment regions of an image given a *templated phrase*. As seen in Figure 1, this requires connecting natural language concepts to image regions. Our experiments shows that the task is challenging for state-of-the-art referring approaches such as *MattNet* [40] and *RMI* [21]. We find that the overall performance is limited by the performance on rare categories and attributes. To address these challenges we present (i) a modular approach for combining visual cues related to categories, attributes, and relationships, and (ii) a systematic approach to improving the performance on rare categories and attributes by leveraging predictions on more frequent ones. Our category and attribute modules are based on detection models, whose instance-level scores are projected back to the image and further processed using an attention-based model driven by the query phrase. Finally, these are combined with

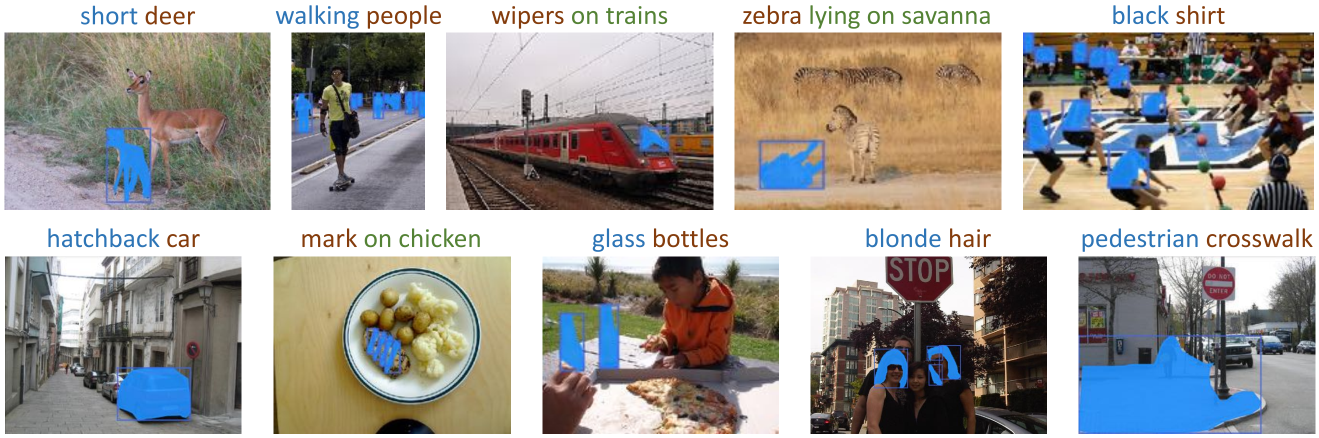

Figure 2. **Example annotations from the VGPhraseCut dataset**. Colors (blue, red, green) of the input phrases correspond to words that indicate attributes, categories, and relationships respectively.

relationship scores to estimate the segmentation mask (see Figure 1). Objects and stuff categories are processed in a unified manner and the modular design, after the treatment of rare categories, outperforms existing end-to-end models trained on the same dataset.

Using the dataset we present a systematic analysis of the performance of the models on different subsets of the data. The main conclusions are: (i) object and attribute detection remains poor on rare and small-sized categories, (ii) for the task of image grounding, rare concepts benefit from related but frequent ones (*e.g*., the concept "policeman" could be replaced by "man" if there were other distinguishing attributes such as the color of the shirt), and (iii) attributes and relationship models provide the most improvements on rare and small-sized categories. The performance on this dataset is far from perfect and should encourage better models of object detection and semantic segmentation in the computer vision community. The dataset and code is available at: `https://people.cs.umass.edu/~chenyun/phrasecut`.

2. Related Work

The language and vision community has put significant effort into relating words and images. Our dataset is closely related to datasets for the visual grounding of referring expressions. We also describe recent approaches for grounding natural language to image regions.

Visual grounding datasets Table 1 shows a comparison of various datasets related to grounding referring expressions to images. The *ReferIt* dataset [17] was collected on images from ImageCLEF using a ReferItGame between two players. Mao *et al*. [26] used the same strategy to collect a significantly larger dataset called *Google RefExp*, on images from the MS COCO dataset [20]. The referring phrases describe objects and refer to boxes inside the image across 80 categories, but the descriptions are long and perhaps redundant. Yu *et al*. [41] instead collect referring expressions using a pragmatic setting where there is limited interaction time between the players to generate and infer the referring object. They collected two versions of the data: *RefCOCO* that allows location descriptions such as "man on the left", and *RefCOCO+* which forbids location cues forcing a focus on other visual clues. One drawback is that *Google RefExp*, *RefCOCO* and *RefCOCO+* are all collected on *MS-COCO* objects, limiting their referring targets to 80 object categories. Moreover, the target is always one single instance, and there is no treatment of stuff categories.

Another related dataset is the *Flickr30K Entities* [30]. Firstly entities are mined and grouped (co-reference resolution) from captions by linking phrases that describe the same entity and then the corresponding bounding-boxes are collected. Sentence context is often needed to ground the entity phrases to image regions. While there are a large number of categories (44,518), most of them have very few examples (average 6.2 examples per category) with a significant bias towards human-related categories (their top 7 categories are "man","woman", "people", "shirt", "girl", "boy", "men"). The dataset also does not contain segmentation masks. nor phrases that describe multiple instances.

Our dataset is based on the *Visual Genome (VG)* dataset [18]. VG annotates each image as a "scene graph" linking descriptions of individual objects, attributes, and their relationships to other objects in the image. The dataset is diverse, capturing various object and stuff categories, as well as attribute and relationship types. However, most descriptions do not distinguish one object from other objects in the scene, *i.e*., they are not referring expressions. Also, VG object boxes are very noisy. We propose a procedure to mine descriptions within the scene graph that uniquely identifies the objects, thereby generating phrases that are more suitable for the referring task. Finally, we collect segmentation annotations of corresponding regions for these phrases.

Dataset	ReferIt [17]	Google RefExp [26]	RefCOCO [41]	Flickr30K Entities [30]	Visual Genome [18]	VGPhraseCut
# images	19,894	26,711	19,994	31,783	108,077	77,262
# instances	96,654	54,822	50,000	275,775	1,366,673	345,486
# categories	-	80	80	44,518	80,138	3103
multi-instance	No	No	No	No	No	Yes
segmentation	Yes	Yes	Yes	No	No	Yes
referring phrase	short phrases	long descriptions	short phrases	entities in captions	region descriptions	templated phrases

Table 1. **Comparison of visual grounding datasets.** The proposed VGPhraseCut dataset has a significantly higher number of categories than RefCOCO and Google RefExp, while also containing multiple instances.

Approaches for grounding language to images Techniques for localizing regions in an image given a natural language phrase can be broadly categorized into two groups: single-stage segmentation-based techniques and two-stage detection-and-ranking based techniques.

Single-stage methods [6, 15, 19, 21, 27, 33, 38, 39] predict a segmentation mask given a natural language phrase by leveraging techniques used in semantic segmentation. These methods condition a feed-forward segmentation network, such as a fully-convolutional network or U-Net, on the encoding of the natural language (*e.g.*, LSTM over words). The advantage is that these methods can be directly optimized for the segmentation performance and can easily handle stuff categories as well as different numbers of target regions. However, they are not as competitive on small-sized objects. We compare a strong baseline of RMI [21] on our dataset.

More state-of-the-art methods are based on a two-stage framework of region proposal and ranking. Significant innovations in techniques have been due to the improved techniques for object detection (*e.g.*, Mask R-CNN [11]) as well as language comprehension. Some earlier works [7, 16, 23, 25, 26, 28, 29, 31, 34, 41] adopt a joint image-language embedding model to rank object proposals according to their matching scores to the input expressions. More recent works improve the proposal generation [7, 42], introduce attention mechanisms [1, 9, 39] for accurate grounding, or leverage week supervision from captions [8, 36].

The two-stage framework has also been further extended to modular comprehension inspired by neural module networks [2]. For example, Hu *et al.* [14] introduce a compositional modular network for better handling of attributes and relationships. Yu *et al.* [40] propose a modular attention network (MattNet) to factorize the referring task into separate ones for the noun phrase, location, and relationships. Liu *et al.* [24] improves MattNet by removing easy and dominant words and regions to learn more challenging alignments. Several recent works [3,4,10,22,35,37,43] also apply reasoning on graphs or trees for more complicated phrases. These approaches have several appealing properties such as more detailed modeling of different aspects of language descriptions. However, these techniques have been primarily evaluated on datasets with a closed set of categories, and often with ground-truth instances provided.

Sadhu *et al.* [32] proposes zero-shot grounding to handle phrases with unseen nouns. Our work emphasizes further on the large number of categories, attributes and relationships, providing supervision over these long-tailed concepts and more detailed and straightforward evaluation.

3. The VGPhraseCut Dataset

In this section, we describe how the VGPhraseCut dataset was collected, the statistics of the final annotations, and the evaluation metrics. Our annotations are based on images and scene-graph annotations from the *Visual Genome (VG)* dataset. We briefly describe each step in the data-collection pipeline illustrated in Figure 3, deferring to the supplemental material Section 1.1 for more details.

Step 1: Box sampling Each image in VG dataset contains 35 boxes on average, but they are highly redundant. We sample an average of 5 boxes from each image in a stratified manner by avoiding boxes that are highly overlapping or are from a category that already has a high number of selected boxes. We also remove boxes that are less than 2% or greater than 90% of the image size.

Step 2: Phrase generation Each sampled box has *several* annotations of category names (*e.g.*, "man" and "person"), attributes (*e.g.*, "tall" and "standing") and relationships with other entities in the image (*e.g.*, "next to a tree" and "wearing a red shirt"). We generate one phrase for one box at a time, by adding categories, attributes and relationships that allow discrimination with respect to other VG boxes by the following set of heuristics:

1. We first examine if one of the provided categories of the selected box is unique. If so we add this to the phrase and tack on to it a randomly sampled attribute or relationship description of the box. The category name uniquely identifies the box in this image.
2. If the box is *not* unique in terms of any of its category names, we look for a unique attribute of the box that distinguishes it from boxes of the same category. If such an attribute exists we combine it with the category name as the generated phrase.
3. If *no* such an attribute exists, we look for a distinguishing relationship description (a relationship predicate plus a category name for the supporting object). If such a relationship exists we combine it with the category name as the generated phrase.

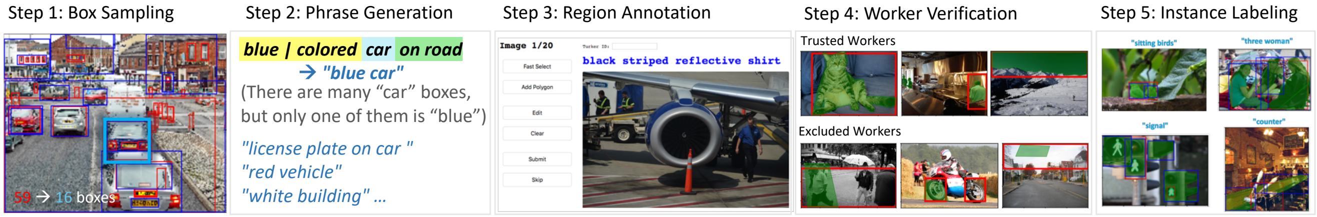

Figure 3. **Illustrations of our VGPhraseCut dataset collection pipeline. Step 1:** blue boxes are the sampling result; red boxes are ignored. **Step 2:** Phrase generation example in the previous image. **Step 3:** User interface for collecting region masks. **Step 4:** Example annotations from trusted and excluded annotators. **Step 5:** Instance label refinement examples. Blue boxes are final instance boxes, and red boxes are corresponding ones from Visual Genome annotations.

4. If all of the above fail, we combine all attributes and relationships on the target box and randomly choose a category from the provided list of categories for the box to formulate the phrase. In this case, the generated phrase is more likely to correspond to more than one instance within the image.

The attribute and relationship information may be missing if the original box does not have any, but there is always a category name for each box. Phrases generated in this manner tend to be concise but do not always refer to a unique instance in the image.

Step 3: Region annotation We present the images and generated phrases from the previous steps to human annotators on Amazon Mechanical Turk, and ask them to draw polygons around the regions that correspond to provided phrases. Around 10% of phrases are skipped by workers when the phrases are ambiguous.

Step 4: Automatic annotator verification Based on manual inspection over a subset of annotators, we design an automatic mechanism to identify trusted annotators based on the overall agreement of their annotations with the VG boxes. Only annotations from trusted annotators are included in our dataset. 9.27% phrase-region pairs are removed in this step.

Step 5: Automatic instance labeling As a final step we generate instance-level boxes and masks. In most cases, each polygon drawn by the annotators is considered an instance. It is further improved by a set of heuristics to merge multiple polygons into one instance and to split one polygon into several instances leveraging the phrase and VG boxes.

3.1. Dataset statistics

Our final dataset consists of 345,486 phrases across 77,262 images. This roughly covers 70% of the images in Visual Genome. We split the dataset into 310,816 phrases (71,746 images) for training, 20,316 (2,971 images) for validation, and 14,354 (2,545 images) for testing. There is no overlap of COCO trainval images with our test split so that models pre-trained on COCO can be fairly used and evaluated. Figure 4 illustrates several statistics of the dataset. Our dataset contains 1,272 unique category phrases, 593 unique attribute phrases, and 126 relationship phrases with frequency over 20, as seen by the word clouds. Among the distribution of phrases (bottom left bar plot), one can see that 68.2% of the instances can be distinguished by category alone (*category*+), while 11.8% of phrases require some treatment of attributes to distinguish instances (*attributes*+). Object sizes and their frequency vary widely. While most annotations refer to a single instance, 17.6% of phrases refer to two or more instances. These aspects of the dataset make the *PhraseCut* task challenging. In Supplementary Section 1.2, we further demonstrate the long-tailed distribution of concepts and how attributes and relationships vary in different categories.

3.2. Evaluation metrics

The *PhraseCut* task is to generate a binary segmentation of the input image given a referring phrase. We assume that the input phrase is parsed into attribute, category, and relationship descriptions. For evaluation we use the following intersection-over-union (IoU) metrics:

- cumulative IoU: $\texttt{cum-IoU} = (\sum_t I_t) \,/\, (\sum_t U_t)$, and
- mean IoU: $\texttt{mean-IoU} = \frac{1}{N}\sum_t I_t/U_t$.

Here t indexes over the phrase-region pairs in the evaluation set, I_t and U_t are the intersection and union area between predicted and ground-truth regions, and N is the size of the evaluation set. Notice that, unlike cum-IoU, mean-IoU averages the performance across all image-region pairs and thus balances the performance on small and large objects.

We also report the precision when each phrase-region task is considered correct if the IoU is above a threshold. We report results with IoU thresholds at 0.5, 0.7, 0.9 as Pr@0.5, Pr@0.7, Pr@0.9 respectively.

All these metrics can be computed on different subsets of the data to obtain a better understanding of the strengths and failure modes of the model.

4. A Modular Approach to PhraseCut

We propose **H**ierarchical Mod**ul**ar **A**ttention **Net**work (HULANet) for the PhraseCut task, as illustrated in Figure 5. The approach is based on two design principles. First,

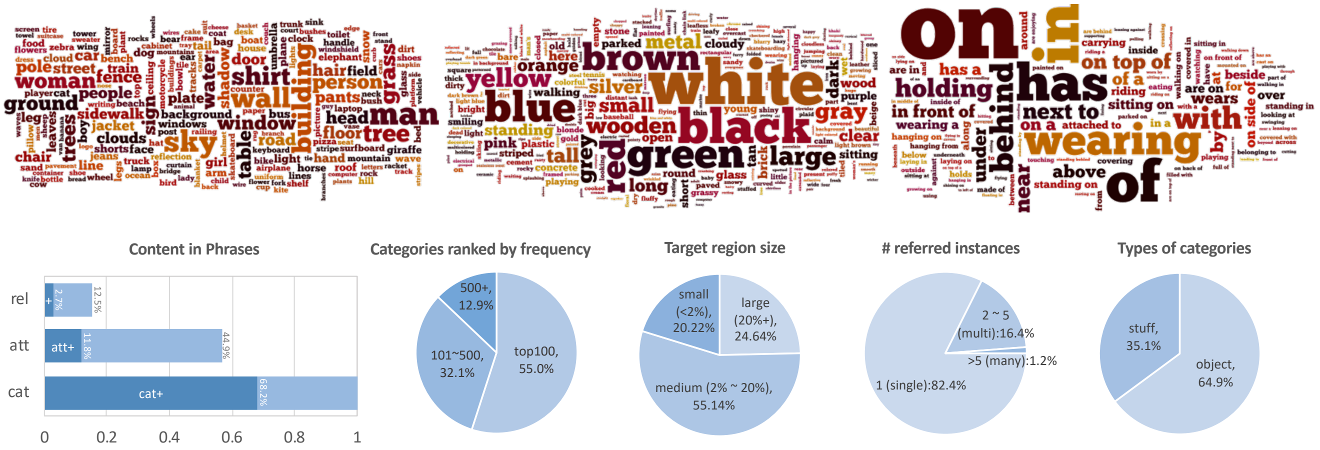

Figure 4. **Statistics of the VGPhraseCut dataset**. **Top row:** Word clouds of categories (left), attributes (center), and relationship descriptions (right) in the dataset. The size of each phrase is proportional to the square root of its frequency in the dataset. **Bottom row:** breakdowns of the dataset into different subsets including contents in phrases (first), category frequency (second), size of target region relative to the image size (third), number of target instances per query phrase (fourth), and types of category (last). The leftmost bar chart shows the breakdown of phrases into those that have category annotation (cat) and those that can be distinguished by category information alone (cat+), and similarly for attributes and relationships.

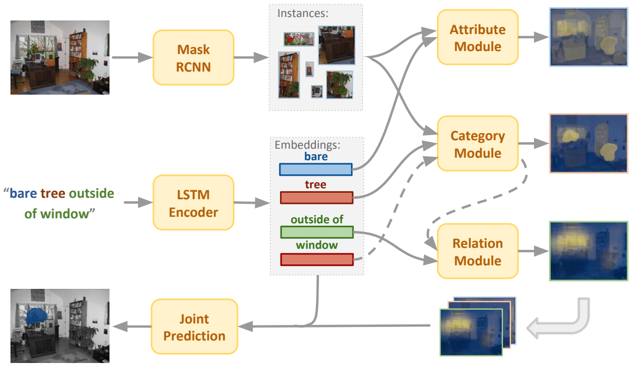

Figure 5. **Architecture of HULANet**. The architecture consists of modules to obtain attribute, category, and relation predictions given a phrase and an image. The attribute and category scores are obtained from Mask-RCNN detections and projected back to the image. The scores across categories and attributes are combined using a module-specific attention model. The relationship module is a convolutional network that takes as input the prediction mask of the related category and outputs a spatial mask given the relationship predicate. The modules are activated based on their presence in the query phrase and combined using an attention mechanism guided by the phrase.

we design individual modules for category, attribute and relationship sub-phrases. Each module handles the long-tail distribution of concepts by learning to aggregate information across concepts using a module-specific attention mechanism. Second, instance-specific predictions are projected onto the image space and combined using an attention mechanism driven by the input phrase. This allows the model to handle stuff and object categories, as well as multiple instances in a unified manner. Details of each module are described next.

Backbone encoders We use the Mask-RCNN [11] detector and bi-directional LSTMs [13] as our backbone encoders for images and phrases respectively. The Mask-RCNN (with ResNet101 [12] backbone) is trained to detect instances and predict category scores for the 1,272 categories that have a frequency over 20 on our dataset. Different from instance detection tasks on standard benchmarks, we allow relatively noisy instance detections by setting a low threshold on objectness scores and by allowing at most 100 detections per image to obtain a high recall. For phrase encoding, we train three separate bi-directional LSTMs to generate embeddings for categories, attributes and relationship phrases. They share the same word embeddings initialized from FastText [5] as the input to the LSTM, and have mean pooling applied on the LSTM output of the corresponding words as the encoded output.

Category module The category module takes as input the phrase embedding of the category and detected instance boxes (with masks) from Mask-RCNN, and outputs a score-map of corresponding regions in the image. We first construct the category channels $C \in \mathbb{R}^{N \times H \times W}$ by projecting the Mask-RCNN predictions back to the image. Here $N = 1272$ is the number of categories and $H \times W$ is set to $1/4\times$ the input image size. Concretely, for each instance i detected by Mask R-CNN as category c_i with score s_i, we project its predicted segmentation mask to image as a binary mask $m_{i,H\times W}$, and update the category channel score at the corresponding location as $C[c_i, m_i] := \max(s_i, C[c_i, m_i])$. Finally, each category channel is passed though a "layer-norm" which scales the mean and variance of each channel.

To compute the attention over the category channels, the phrase embedding e_{cat} is passed through a few linear layers f with sigmoid activation at the end to predict the attention weights over the category channels $A = \sigma(f(e_{cat}))$. We calculate the weighted sum of the category channels guided by the attention weights $S_{H \times W} = \sum_c A_c \cdot C_c$, and apply a learned affine transformation plus sigmoid to obtain the category module prediction heat-map $P_{H \times W} = \sigma(a \cdot S_{H \times W} + b)$. This attention scheme enables the category module to leverage predictions from good category detectors to improve performance on more difficult categories. We present other baselines for combining category scores in the ablation studies in Section 5.

Attribute module The attribute module is similar to the category module except for an extra attribute classifier. On top of the pooled ResNet instance features from Mask-RCNN, we train a two-layer multi-label attribute classifier. To account for significant label imbalance we weigh the positive instances more when training attribute classifiers with the binary cross-entropy loss. To obtain attribute score channels we take the top 100 detections and project their top 20 predicted attributes back to the image. Identical with the category module, we use the instance masks from the Mask-RCNN, update the corresponding channels with the predicted attribute scores, and finally apply the attention scheme guided by the attribute embedding from the phrase to obtain the final attribute prediction score heat-map.

Relationship module Our simple relationship module uses the category module to predict the locations of the supporting object. The down-scaled (32×32) score of the supporting object is concatenated with the embedding of the relationship predicate. This is followed by two dilated convolutional layers with kernel size 7 applied on top, achieving a large receptive field without requiring many parameters. Finally, we apply an affine transformation followed by sigmoid to obtain the relationship prediction scores. The convolutional network can model coarse spatial relationships by learning filters corresponding to each spatial relation. For example, by dilating the mask one can model the relationship "near", and by moving the mask above one can model the relationship "on".

Model	mean-IoU	cum-IoU	Pr@0.5	Pr@0.7	Pr@0.9
HULANet					
cat	39.9	48.8	40.8	25.9	5.5
cat+att	**41.3**	**50.8**	**42.9**	**27.8**	**5.9**
cat+rel	41.1	49.9	42.3	26.6	5.6
cat+att+rel	**41.3**	50.2	42.4	27.0	5.7
Mask-RCNN self	36.2	45.9	37.2	22.9	4.1
Mask-RCNN top	39.4	47.4	40.9	25.8	4.8
RMI	21.1	42.5	22.0	11.6	1.5
MattNet	20.2	22.7	19.7	13.5	3.0

Table 2. **Comparison of various approaches on the entire test set of VGPhraseCut.** We compare different combinations of modules in our approach (HULANet) against baseline approaches: Mask-RCNN, RMI and MattNet.

Combining the modules The category, attribute, and relation scores P_c, P_a, P_r obtained from individual modules are each represented as a $H \times W$ image, $1/4$ the image size. To this we append channels of quadratic interactions $P_i \circ P_j$ for every pair of channels (including $i = j$), obtained using elementwise product and normalization, and a bias channel of all ones, to obtain a 10-channel scoremap F (3+6+1 channels). Phrase embeddings of category, attribute and relationship are concatenated together and then encoded into 10-dimensional "attention" weights w through linear layers with LeakyReLU and DropOut followed by normalization. When there is no attribute or relationship in the input phrase, the corresponding attention weights are set to zero and the attention weights are re-normalized to sum up to one. The overall prediction is the attention-weighted sum of the linear and quadratic feature interactions: $O = \sum_t F_t w_t$. Our experiments show a slight improvement of 0.05% on validation `mean-IoU` with the quadratic features.

Training details The Mask-RCNN is initialized with weights pre-trained on the MS-COCO dataset [20] and fine-tuned on our dataset. It is then fixed for all the experiments. The attribute classifier is trained on ground-truth instances and their box features pooled from Mask-RCNN with a binary cross-entropy loss specially weighted according to attribute frequency. These are also fixed during the training of the referring modules. On top of the fixed Mask-RCNN and the attribute classifier, we separately train the individual category and attribute modules. When combining the modules we initialize the weights from individual ones and fine-tune the whole model end-to-end. We apply a pixel-wise binary cross-entropy loss on the prediction score heat-map from each module and also on the final prediction heat-map. To account for the evaluation metric (`mean-IoU`), we increase the weights on the positive pixels and average the loss over referring phrase-image pairs instead of over pixels. All our models are trained on the training set. For evaluation, we require a binary segmentation mask which is obtained by thresholding on prediction scores. These thresholds are set based on `mean-IoU` scores on the validation set. In the next section, we report results on the test set.

5. Results and Analysis

5.1. Comparison to baselines

Table 2 shows the overall performance of our model and its ablated versions with two baselines: RMI [21] and MattNet [40]. They yield near state-of-the-art performance on datasets such as RefCOCO [17].

RMI is a single-stage visual grounding method. It extracts spatial image features through a convolutional encoder, introduces convolutional multi-modal LSTM for jointly modeling of visual and language clues in the bottleneck, and predicts the segmentation through an upsampling

Model	all	coco	1-100	101-500	500+
HULANet					
cat	39.9	46.5	46.8	31.8	25.2
cat+att	**41.3**	**48.3**	**48.2**	33.6	26.6
cat+rel	41.1	47.9	47.8	33.6	26.6
cat+att+rel	**41.3**	47.8	47.8	**33.8**	**27.1**
Mask-RCNN self	36.2	44.9	45.5	27.9	10.1
Mask-RCNN top	39.4	46.1	46.4	31.6	23.2
RMI	21.1	23.7	28.4	12.7	5.5
MattNet	20.2	19.3	24.9	14.8	10.6

Table 3. **The `mean-IoU` on VGPhraseCut test set for various category subsets.** The column *coco* refers to the subset of data corresponding to the 80 coco categories, while the remaining columns show the performance on the top 100, 101-500 and 500+ categories in the dataset sorted by frequency.

Model	all	att	att+	rel	rel+	stuff	obj
HULANet							
cat	39.9	37.6	37.4	32.3	33.0	47.2	33.9
cat+att	**41.3**	**39.1**	**38.8**	33.7	33.8	**48.4**	35.5
cat+rel	41.1	38.8	38.4	33.8	**34.0**	48.1	35.4
cat+att+rel	**41.3**	39.0	38.5	**34.1**	33.9	48.3	**35.6**
Mask-RCNN self	36.2	34.5	34.7	29.0	30.8	44.4	29.5
Mask-RCNN top	39.4	37.3	36.6	31.9	32.6	46.4	33.6
RMI	21.1	19.0	21.0	11.6	12.2	31.1	13.0
MattNet	20.2	19.0	18.9	15.6	15.1	25.5	16.0
Model	all	single	multi	many	small	mid	large
HULANet							
cat	39.9	41.2	37.0	34.3	15.1	40.3	67.6
cat+att	**41.3**	**42.6**	**38.6**	**35.9**	17.1	**42.0**	68.0
cat+rel	41.1	42.5	38.2	35.5	17.1	41.5	**68.2**
cat+att+rel	**41.3**	**42.6**	38.4	35.7	17.3	41.7	**68.2**
Mask-RCNN self	36.2	37.2	34.1	29.9	17.0	35.7	59.4
Mask-RCNN top	39.4	40.6	36.8	33.4	**18.5**	39.3	63.6
RMI	21.1	23.1	16.9	12.7	1.2	18.6	49.5
MattNet	20.2	22.2	15.9	12.6	6.1	18.9	39.5

Table 4. **The `mean-IoU` on VGPhraseCut test set for additional subsets.** *att/rel*: the subset with attributes/relationship annotations; *att+/rel+*: the subset which requires attributes or relationships to distinguish the target from other instances of the same category; *single/multi/many*: subsets that contain different number of instances referred by a phrase; *small/mid/large*: subsets with different sizes of the target region.

decoder. We use the RMI model with ResNet101 [12] as the image encoder. We initialized the ResNet with weights pretrained on COCO [20], trained the whole RMI model on our training data of image region and referring phrase pairs following the default setting as in their public repository, and finally evaluated it on our test set.

RMI obtains high `cum-IoU` but low `mean-IoU` scores because it handles large targets well but fails on small ones (see Table 4 "small/mid/large" subsets). `cum-IoU` is dominated by large targets while our dataset many small targets: 20.2% of our data has the target region smaller than 2% of the image area, while the smallest target in RefCOCO is 2.4% of the image. Figure 6 also shows that RMI predicts empty masks on challenging phrases and small targets.

MattNet focuses on ranking the referred box among candidate boxes. Given a box and a phrase, it calculates the subject, location, and relationship matching scores with three separate modules, and predicts attention weights over the three modules based on the input phrase. Finally, the three scores are combined with weights to produce an overall matching score, and the box with the highest score is picked as the referred box.

We follow the training and evaluation setup described in their paper. We train the Mask-RCNN detector on our dataset, and also train MattNet to pick the target instance box among ground-truth instance boxes in the image. Note that MattNet training relies on complete annotations of object instances in an image, which are used not only as the candidate boxes but also as the context for further reasoning. The objects in our dataset are only sparsely annotated, hence we leverage the Visual Genome boxes instead as context boxes. At test time the top 50 Mask-RCNN detections from all categories are used as input to the MattNet model.

While this setup works well on RefCOCO, it is problematic on VGPhraseCut because detection is more challenging in the presence of thousands of object categories. MattNet is able to achieve `mean-IoU` = 42.4% when the ground-truth instance boxes are provided in evaluation, but its performance drops to `mean-IoU` = 20.2% when Mask-RCNN detections are provided instead. If we only input the detections of the referred category to MattNet, `mean-IoU` improves to 34.7%, approaching the performance of *Mask-RCNN self*, but it still performs poorly on rare categories.

Our modular approach for computing robust category scores from noisy detections alone (*HULANet cat*) outperforms both baselines by a significant margin. Example results using various approaches are shown in Figure 6. Heatmaps from submodules and analysis of failure cases are included in Supplemental Section 3.

5.2. Ablation studies and analysis

Table 3 shows that the performance is lower for rare categories. Detection of thousands of categories is challenging, but required to support open-vocabulary natural language descriptions. However, natural language is also redundant. In this section we explore if a category can leverage scores from related categories to improve performance, especially when it is rare.

First we evaluate Mask-RCNN as a detector, by using the mask of the top-1 detected instance from the referred category as the predicted region. The result is shown as the row *"Mask-RCNN self"* in Table 3. The row below *"Mask-RCNN top"* shows the performance of the model where each category is matched to a single other category based on the best `mean-IoU` on the training set. For example, a category "pedestrian" may be matched to "person" if the person detector is more reliable. Supplemental Section 2 shows the full matching between source and target categories. As one can see in Table 3, the performance on the tail categories jumps significantly (10.1% → 23.2% on the 500+ subset.) In general the tail category detectors are poor

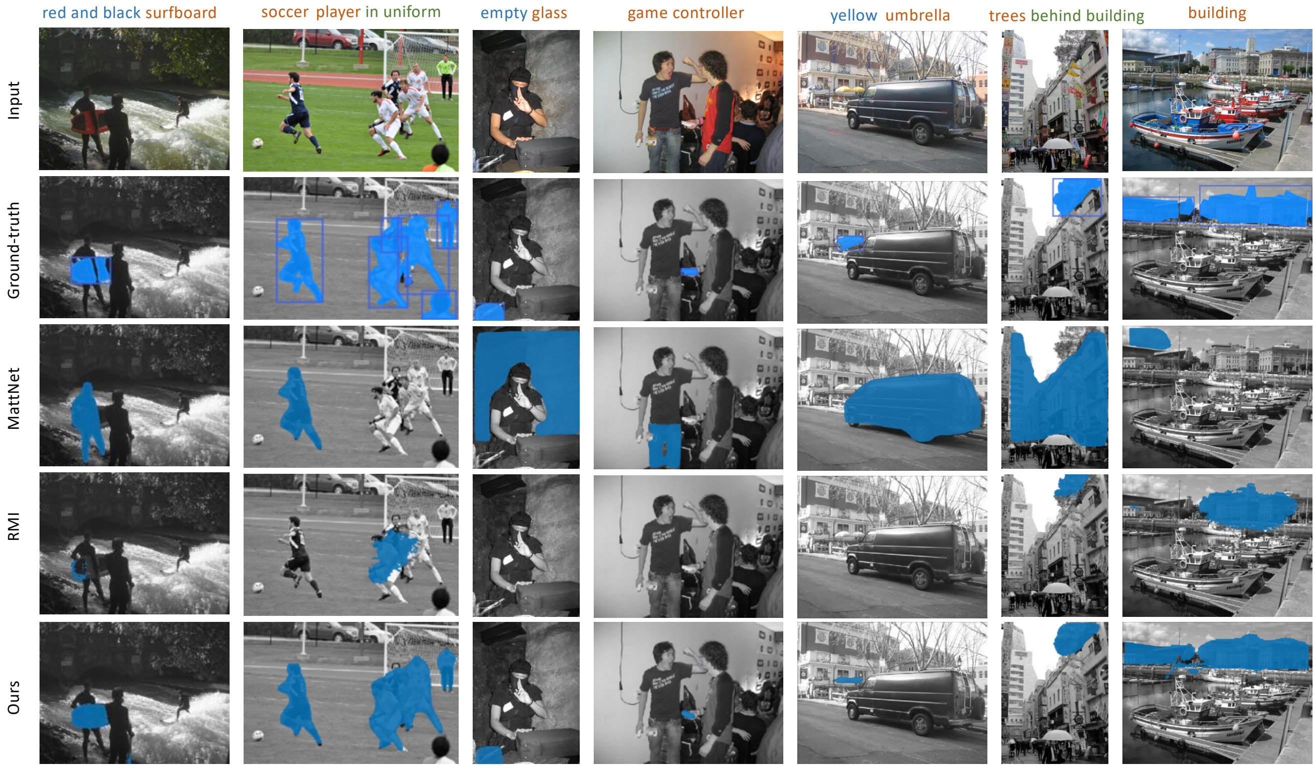

Figure 6. **Prediction results on VGPhraseCut dataset.** Rows from top to down are: (1) input image; (2) ground-truth segmentation and instance boxes; (3) MattNet baseline; (4) RMI baseline; (5) HULANet (cat + att + rel). See more results in the supplemental material.

and rarely used. This also points to a curious phenomenon in referring expression tasks where even though the named category is specific, one can get away with a coarse category detector. For example, if different animal species never appear together in an image, one can get away with a generic animal detector to resolve any animal species.

This also explains the performance of the category module with the category-level attention mechanism. Compared to the single category picked by the Mask-RCNN top model, the ability of aggregating multiple category scores using the attention model provides further improvements for the tail categories. Although not included here, we find a similar phenomenon with attributes, where a small number of base attributes can support a larger, heavy-tailed distribution over the attribute phrases. It is reassuring that the number of visual concepts to be learned grows sub-linearly with the number of language concepts. However, the problem is far from solved as the performance on tail categories is still significantly lower.

Table 4 shows the results on additional subsets of the test data. Some high-level observations are that: (i) Object categories are more difficult than stuff categories. (ii) Small objects are extremely difficult. (iii) Attributes and relationships provide consistent improvements across different subsets. Remarkably, the improvements from attributes and relationships are more significant on rare categories and small target regions where the category module is less accurate.

6. Conclusion

We presented a new dataset, VGPhraseCut, to study the problem of grounding natural language phrases to image regions. By scaling the number of categories, attributes, and relations we found that existing approaches that rely on high-quality object detections show a dramatic reduction in performance. Our proposed HULANet performs significantly better, suggesting that dealing with long-tail object categories via modeling their relationship to other categories, attributes, and spatial relations is a promising direction of research. Another take away is that decoupling representation learning and modeling long-tails might allow us to scale object detectors to rare categories, without requiring significant amount of labelled visual data. Nevertheless, the performance of the proposed approach is still significantly below human performance which should encourage better modeling of language and vision.

Acknowledgements The project is supported in part by NSF Grants 1749833 and 1617917, and Faculty awards from Adobe. Our experiments were performed on the UMass GPU cluster obtained under the Collaborative Fund managed by the Massachusetts Technology Collaborative.

References

[1] Hassan Akbari, Svebor Karaman, Surabhi Bhargava, Brian Chen, Carl Vondrick, and Shih-Fu Chang. Multi-level multimodal common semantic space for image-phrase grounding. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. GraphGround: Graph-Based Language Grounding. In *International Conference on Computer Vision (ICCV)*, 2019.

[4] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. GraphGround: Graph-Based Language Grounding. In *International Conference on Computer Vision (ICCV)*, 2019.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[6] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *International Conference on Computer Vision (ICCV)*, 2019.

[7] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *International Conference on Computer Vision (ICCV)*, 2017.

[8] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. Align2ground: Weakly supervised phrase grounding guided by image-caption alignment. In *International Conference on Computer Vision (ICCV)*, 2019.

[9] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (seqGROUND). In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *International Conference on Computer Vision ICCV*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision (ECCV)*, 2016.

[16] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Empirical methods in natural language processing (EMNLP)*, 2014.

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[19] Ruiyu Li, Kai-Can Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[21] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *International Conference on Computer Vision (ICCV)*, 2017.

[22] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *International Conference on Computer Vision (ICCV)*, 2019.

[23] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *International Conference on Computer Vision (ICCV)*, 2017.

[24] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[26] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *European Conference on Computer Vision (ECCV)*, 2018.

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

[29] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *European Conference on Computer Vision (ECCV)*, 2018.

[30] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123(1):74–93, 2017.

[31] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.

[32] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *International Conference on Computer Vision (ICCV)*, 2019.

[33] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[34] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *International Conference on Computer Vision (ICCV)*, 2019.

[38] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *International Conference on Computer Vision (ICCV)*, 2019.

[39] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. MattNet: Modular attention network for referring expression comprehension. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[41] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, 2016.

[42] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. *CoRR*, abs/1805.03508, 2018.

[43] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.