Segmentation from Natural Language Expressions

Ronghang Hu[1(✉)], Marcus Rohrbach[1,2], and Trevor Darrell[1]

[1] UC Berkeley EECS, Berkeley, CA, USA
{ronghang,rohrbach,trevor}@eecs.berkeley.edu
[2] ICSI, Berkeley, CA, USA

Abstract. In this paper we approach the novel problem of segmenting an image based on a natural language expression. This is different from traditional semantic segmentation over a predefined set of semantic classes, as e.g., the phrase *"two men sitting on the right bench"* requires segmenting only the two people on the right bench and no one standing or sitting on another bench. Previous approaches suitable for this task were limited to a fixed set of categories and/or rectangular regions. To produce pixelwise segmentation for the language expression, we propose an end-to-end trainable recurrent and convolutional network model that jointly learns to process visual and linguistic information. In our model, a recurrent neural network is used to encode the referential expression into a vector representation, and a fully convolutional network is used to a extract a spatial feature map from the image and output a spatial response map for the target object. We demonstrate on a benchmark dataset that our model can produce quality segmentation output from the natural language expression, and outperforms baseline methods by a large margin.

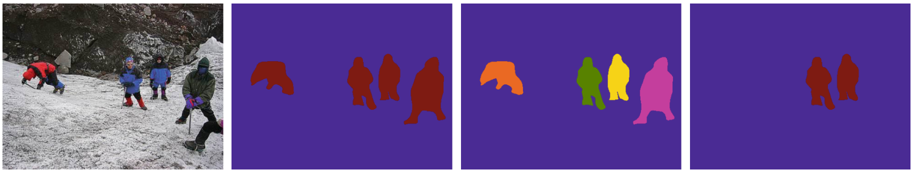
Keywords: Natural language · Segmentation · Recurrent neural network · Fully convolutional network

1 Introduction

Semantic image segmentation is a core problem in computer vision and significant progress has been made using large visual datasets and rich representations based on convolution neural networks [4,6,17,21,32,33]. Although these existing segmentation methods can predict precise pixelwise masks for query categories like "train" or "cat", they are not capable of predicting segmentation for more complicated queries such as the natural language expression "the two people on the right side of the car wearing black shirts".

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46448-0_7) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part I, LNCS 9905, pp. 108–124, 2016.
DOI: 10.1007/978-3-319-46448-0_7



(a) input image (b) object class segmentation of class ***people*** (c) object instance segmentation of class ***people*** (d) segmentation from expression *"people in blue coat"*

Fig. 1. In this work we approach the novel problem of *segmentation from natural language expressions*, which is different from traditional semantic image segmentation and object instance segmentation, as visualized in this figure. (Color figure online)

In this paper we address the following problem: given an image and a natural language expression that describes a certain part of the image, we want to segment the corresponding region(s) that covers the visual entities described by the expression. For example, as shown in Fig. 1(d), for the phrase e.g. *"people in blue coat"* we want to predict a segmentation that covers the two people in the middle wearing blue coat, but not the other two people. This problem is related to but different from the core computer vision problems of *semantic segmentation* (e.g. PASCAL VOC segmentation challenge on 20 object classes [10]), which is concerned with predicting the pixelwise label for a predefined set of object or stuff categories (Fig. 1b), and *instance segmentation* (e.g. [12]), which additionally distinguishes different instances of an object class (Fig. 1c). It also differs from language-independent foreground segmentation (e.g. [24]), where the goal is to generate a mask over the foreground (or the most salient) object. Instead of assigning a semantic label to every pixel in the image as in semantic image segmentation, the goal in this paper is to produce a segmentation mask for the visual entities of interest based on the given expression. Rather than being fixed on a set of object and stuff categories, natural language descriptions may involve also attributes such as *"black"* and *"smooth"*, actions such as *"running"*, spatial relationships such as *"on the right"* and interactions between different visual entities such as *"the person who is riding a horse"*.

The task of segmenting an image from natural language expressions has a wide range of applications, such as building language-based human-robot interface to give instructions like *"pick up the jar on the table next to the apples"*. Here, it is important to be able to use multi-word referential expressions to distinguish between different object instances but also important to get a precise segmentation in contrast to just a bounding box, especially for non-grid-aligned objects (see e.g. Fig. 2). This could also be interesting for interactive photo editing where one could refer with natural language to certain parts or objects of the image to be manipulated, e.g. *"blur the person with a red shirt"*, or referring to parts of your meal to estimate their nutrition, *"two large pieces of bacon"*, to decide better if one should eat it rather than the full meal as in [20].

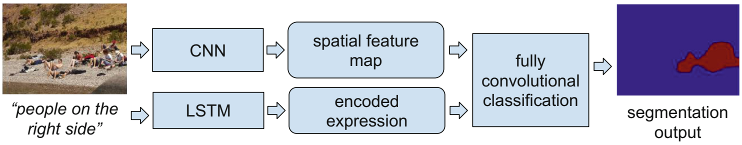


Fig. 2. Overview of our method for segmentation from natural language expressions.

As described in more details in Sect. 2, prior methods suitable for this task were limited to resolving only a bounding box in the image [15,18,23], and/or were limited to a fixed set of categories determined *a priori* [6,17,32,33]. In this paper, we propose an end-to-end trainable recurrent convolutional network model that jointly learns to process visual and linguistic information, and produces segmentation output for the target image region described by the natural language expression, as illustrated in Fig. 2. We encode the expression into a fixed-length vector representation through a recurrent Long short-term memory network (LSTM), and use a convolutional neural network (CNN) to extract a spatial feature map from the image. The encoded expression and the feature map are then processed by a multi-layer classifier network in a fully convolutional manner to produce a coarse response map, which is upsampled with deconvolution [17,21] to obtain a pixel-level segmentation mask of the target image region. Experimental results demonstrate that our model can generate quality segmentation predictions from natural language expressions, and outperforms baseline methods significantly. Our model is trained using standard back-propagation, and is much more efficient at test time than previous approaches relying on scoring each bounding box.

2 Related Work

Localizing Objects with Natural Language. Our work is related to recent work on object localization with natural language, where the task is to localize a target object in a scene from its natural language description (by drawing a bounding box over it). The methods reported in [15,18] build upon image captioning frameworks such as LRCN [8] or mRNN [19], and localize objects by selecting the bounding box where the expression has the highest probability. Our model differs from [15,18] in that we do not have to learn to generate expressions from image regions. Rohrbach *et al.* [23] propose a model to localize a textual phrase by attending to a region on which the phrase can be best reconstructed. In [22], Canonical Correlation Analysis (CCA) is used to learn a joint embedding space of visual features and words, and given a natural language query, the corresponding target object is localized by finding the closest region to the text sequence in the joint embedding space. Also, the concept of visual phrases [26] is related to our work as it captures compositions of multiple words or objects. However, [26] only deals with 17 manually chosen object compositions, whereas our method captures much richer queries represented by natural language.

To the best of our knowledge, all previous localization methods can only return a bounding box of the target object, and no prior work has learned to directly output a segmentation mask of an object given a natural language description as query. As a comparison, in Sect. 4.1 we also evaluate using foreground segmentation over the bounding box prediction from [15,23].

Fully Convolutional Networks for Segmentation. Fully convolutional networks are convolutional neural networks consisting of only convolutional (and pooling) layers, which are the state-of-the-art method for semantic segmentation over a pre-defined set of semantic categories [6,17,32,33]. A nice property of fully convolutional networks is that spatial information is preserved in the output, which makes these networks suitable for segmentation tasks that require spatial grid output. In our model, both feature extraction and segmentation output are performed through fully convolutional networks. We also use a fully convolution network for per-word segmentation as a baseline in Sect. 4.1.

Attention and Visual Question Answering. Recently, attention models have been used in several areas including image recognition, image captioning and visual question answering. In [30], image captions are generated through focusing on a specific image region for each word. In recent visual question answering models [29,31], the answer is determined through attending to one or multiple image regions. Andreas *et al.* [2] propose a visual question answering method that answers object reference questions like *"Where is the black cat?"* by parsing the sentence and generating individual attention maps for *"black"* and *"cat"* and then combining them. This mechanism has some similarity to our per-word baselines.

These attention models are related to our work as they also learn to generate spatial grid "attention maps" which often cover the objects of interest. However, these attention models differ from our work as they only learn to generate coarse spatial outputs and the purpose of the attention map is to facilitate other tasks such as image captioning, rather than a precise segmentation of the object.

3 Our Model

Given an image and a natural language expression as query, the goal is to output a segmentation mask for the visual entities described by the expression. This problem requires both visual and linguistic understanding of the image and the expression. To accomplish this goal, we propose a model with three main components: a natural language expression encoder based on a recurrent LSTM network, a fully convolutional network to extract local image descriptors and generate a spatial feature map, and a fully convolutional classification and upsampling network that takes as input the encoded expression and the spatial feature map and outputs a pixelwise segmentation mask. Figure 3 shows the outline of our method; we introduce the details of these components in Sects. 3.1, 3.2 and 3.3. The network architecture for feature map extraction and classification is similar to the FCN model [17], which has been shown effective for semantic image segmentation.

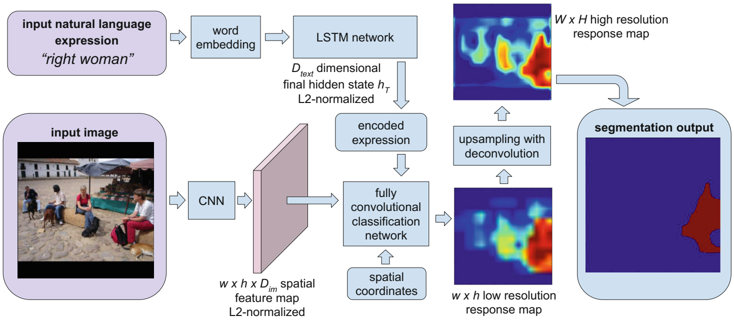

input natural language expression
"right woman"
word embedding
LSTM network
D_{text} dimensional final hidden state h_T L2-normalized
W x H high resolution response map
encoded expression
upsampling with deconvolution
segmentation output
input image
CNN
fully convolutional classification network
spatial coordinates
w x h x D_{im} spatial feature map L2-normalized
w x h low resolution response map

Fig. 3. Our model for segmentation from natural language expressions consists of three main components: an expression encoder based upon a recurrent LSTM network, a fully convolutional network to generate a spatial feature map, and a fully convolutional classification and upsampling network to predict pixelwise segmentation.

Compared with related work [15,18], we do not explicitly produce a word sequence corresponding to object descriptions given a visual representation, since we are interested in predicting image segmentation from an expression rather than predicting the expression. In this way, our model has less parameters compared with [15,18] as it does not have to learn to predict the next word.

3.1 Spatial Feature Map Extraction

Given an image of a scene, we want to obtain a discriminative feature representation of it while preserving the spatial information in the representation so that it is easier to predict a spatial segmentation mask. This is accomplished through a fully convolutional network model similar to FCN-32s [17], where the image is fed through a series of convolutional (and pooling) layers to obtain a spatial map output as feature representation. Given an input image of size $W \times H$, we obtain a $w \times h$ spatial feature map, with each position on the feature map containing D_{im} channels (D_{im} dimensional local descriptors).

For each spatial location on the feature map, we apply L2-normalization to the D_{im} dimensional local descriptor at that position in order to obtain a more robust feature representation. In this way, we can extract a $w \times h \times D_{im}$ spatial feature map as the representation for each image.

Also, to allow the model to reason about spatial relationships such as "right woman" in Fig. 3, two extra channels are added to the feature maps: the x and y coordinate of each spatial location. We use relative coordinates, where the upper left corner and the lower right corner of the feature map are represented as $(-1, -1)$ and $(+1, +1)$, respectively. In this way, we obtain a $w \times h \times (D_{im} + 2)$ representation containing local image descriptors and spatial coordinates.

In our implementation, we adopt the VGG-16 architecture [27] as our fully convolutional network by treating fc6, fc7 and fc8 as convolutional layers, which outputs $D_{im} = 1000$ dimensional local descriptors. The resulting feature map size is $w = W/s$ and $h = H/s$, where $s = 32$ is the pixel stride on fc8 layer output. The units on the spatial feature map have a very large receptive field of 384 pixels, so our method has the potential to aggregate contextual information from nearby regions, which can help to reason about interaction between visual entities, such as "the man next to the table".

3.2 Encoding Expressions with LSTM Network

For the input natural language expression that describes an image region, we would like to represent the text sequence as a vector since it is easier to process fixed-length vectors than variable-length sequences. To achieve this goal, we take the encoder approach in sequence to sequence learning methods [7,28]. In our encoder for the natural language expression, we first embed each word into a vector through a word embedding matrix, and then use a recurrent Long-Short Term Memory (LSTM) [13] network with D_{text} dimensional hidden state to scan through the embedded word sequence. For a text sequence $S = (w_1, ..., w_T)$ with T words (where w_t is the vector embedding for the t-th word), at each time step t, the LSTM network takes as input the embedded word vector w_t from the word embedding matrix. At the final time step $t = T$ after the LSTM network has seen the whole text sequence, we use the hidden state h_T of the LSTM network as the encoded vector representation of the expression. Similar to Sect. 3.1, we also L2-normalize the D_{text} dimensions in h_T. We use an LSTM network with a $D_{text} = 1000$ dimensional hidden state in our implementation.

3.3 Spatial Classification and Upsampling

After extracting the spatial feature map from the image in Sect. 3.1 and the encoded expression h_T in Sect. 3.2, we want to determine whether or not each spatial location on the feature map belongs the foreground (the visual entities described by the natural language expression). In our model, this is done by a fully convolutional classifier over the local image descriptor and the encoded expression. We first tile and concatenate h_T to the local descriptor at each spatial location in the spatial grid to obtain a $w \times h \times D^*$ (where $D^* = D_{im} + D_{text} + 2$) spatial map containing both visual and linguistic features. Then, we train a two-layer classification network, with a $D_{cls} = 500$ dimensional hidden layer, which takes as input the D^* dimensional representation and output a score to indicate whether a spatial location belong to the target image region or not.

This classification network is applied in a fully convolutional way over the underlying $w \times h$ feature map as two 1×1 convolutional layers (with ReLU nonlinearity between them). The fully convolutional classification network outputs a $w \times h$ coarse *low-resolution response map* containing classification scores, which can be seen as a low-resolution segmentation of the referential expression, as shown in Fig. 3.

In order obtain a segmentation mask with higher resolution, we further perform upsampling through deconvolution (swapping the forward and backward pass of convolution operation) [17,21]. Here we use a $2s \times 2s$ deconvolution filter with stride s (where $s = 32$ for the VGG-16 network architecture we use), which is similar to the FCN-32s model [17]. The deconvolution operation produces a $W \times H$ *high resolution response map* that has the same size as the input image, and the values on the high resolution response map represent the confidence of whether a pixel belongs to the target object. We use the pixelwise classification results (i.e. whether a pixel score is above 0) as the final segmentation prediction.

At training time, each training instance in our training set is a tuple (I, S, M), where I is an image, S is a natural language expression describing a region within that image, and M is a binary segmentation mask of that region. The loss function during training is defined as the average over pixelwise loss

$$Loss = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} L(v_{ij}, M_{ij}) \tag{1}$$

where W and H are image width and height, v_{ij} is the response value (score) on the high resolution response map and M_{ij} is the binary ground-truth label at pixel (i, j). L is the per-pixel weighed logistic regression loss as follows

$$L(v_{ij}, M_{ij}) = \begin{cases} \alpha_f \log(1 + \exp(-v_{ij})) & \text{if } M_{ij} = 1 \\ \alpha_b \log(1 + \exp(v_{ij})) & \text{if } M_{ij} = 0 \end{cases} \tag{2}$$

where α_f and α_b are loss weights for foreground and background pixels. In practice, we find that training converges faster using higher loss weights for foreground pixels, and we use $\alpha_f = 3$ and $\alpha_b = 1$ in $L(v_{ij}, M_{ij})$.

The parameters in the feature map extraction network are initialized from a VGG-16 network [27] pretrained on the 1000-class ILSVRC classification task [25], the deconvolution filter for upsampling is initialized from bilinear interpolation. All other parameters in our model, including the word embedding matrix, the LSTM parameters and the classifier parameters, are randomly initialized. The whole network is trained with standard back-propagation using SGD with momentum. Our model is implemented using TensorFlow [1], and our code and data are available at http://ronghanghu.com/text_objseg.

4 Experiments

Compared with the widely used datasets in image segmentation such as PASCAL VOC [10], there are only a few publicly available datasets with natural language annotations over segmented image regions. In our experiments, we train and test our method on the ReferIt dataset [16] with natural language descriptions of visual entities and their segmentation masks. The ReferIt dataset [16] is built upon the IAPR TC-12 dataset [11] and has 20,000 images. There are 130,525 expressions annotated on 96,654 segmented image regions (some regions are annotated with multiple expressions). In this dataset, the ground-truth segmentation comes from the SAIAPR-12 dataset [9]. The expressions in the ReferIt dataset are discriminative for the regions, as they were collected in a two-player game whose goal was to make the target region easily distinguishable through the expression from the rest of the image. At the time of writing, the ReferIt dataset [16] is the biggest publicly available dataset that contains natural language expressions annotated on segmented image regions.

On this dataset, we use the same trainval and test split as in [15,23]. There are 10,000 images for training and validation, and 10,000 images for testing. The annotated regions in the ReferIt dataset contains both "object" regions such as car, person and bottle and "stuff" regions such as sky, river and mountain.

Since there has not been prior work that directly learns to predict segmentation based on natural language expressions as far as we know, to evaluate our method, we construct several strong baseline methods as described in Sect. 4.1, and compare our approach with these methods. All the baselines and our method are trained on the ReferIt dataset for comparison.

4.1 Baseline Methods

Combination of Per-word Segmentation. In this baseline method, instead of first encoding the whole expression with a recurrent LSTM network, each word in the expression is segmented individually, and the per-word segmentation results are then combined to obtain the final prediction. This method can be seen as using a "bag-of-word" representation of the expression. We take the N most frequently appearing words in ReferIt dataset (after manually removing some stop words like "the" and "towards"), and train a FCN model [17] to segment each word. Similar to the PASCAL VOC segmentation challenge [10], in this method, each word is treated as an independent semantic category. However, unlike in PASCAL VOC segmentation, here a pixel can belong to multiple categories (words) simultaneously and thus have multiple labels. During training, we generate a per-word pixelwise label map for each training sample (an image and an expression) in the training set. For a given expression, the corresponding foreground pixels are labeled with a N-dimensional binary vector l, where $l_i = 1$ if and only if word i is present in the expression, and background pixels are labeled with l equal to all zeros. In our experiments, we use $N = 500$ and initialize the network from a FCN-32s network pretrained on PASCAL VOC 2011 segmentation task [17], and train the whole network with a multi-label logistic regression loss over the words.

At test time, given an image and a natural language expression as input, the network outputs pixelwise score maps for the N words, and the per-word scores are further combined to obtain the segmentation for the input expression. In our implementation, we experiment with three different approaches to combine the per-word segmentation: for those words (among the N-word list) that appear in the expression, we (a) take the average of their scores or (b) take the intersection of their prediction or (c) take the union of their prediction. In some rare cases (2.83 % of the test samples), none of the words in the expression are among the N most frequent words, and we do not output any segmentation for this expression, i.e. all pixels are predicted as background.

Foreground Segmentation from Bounding Boxes. In this baseline method, we first use a localization method based on natural language input [15,23] to obtain a bounding box localization of the given expression, and then extract the foreground segmentation from the bounding box using GrabCut [24]. Given an image and a natural language expression, we use two recently proposed methods SCRC [15] and GroundeR [23] to obtain a bounding box prediction from the image and the expression. SCRC uses a model adapted from image captioning and localizes a referential expression by finding the candidate bounding box where the expression receives the highest probability. GroundeR relies on an attention model over candidate bounding boxes to ground (localize) a referential expression, either in an unsupervised manner by finding the region that can best reconstruct the expression, or in a supervised manner to directly train the model to attend to the best bounding box. In this work we use a re-implementation of the fully-supervised GroundeR. Following [15,23], we use 100 top-scoring EdgeBox [34] proposals as a set of candidate bounding boxes for each image. At test time, given an input expression, we compute the scores of the 100 EdgeBox proposals using SCRC [15] or GroundeR [23], and evaluate two approaches: either using the entire rectangular region of the highest scoring bounding box, or the foreground segmentation from it using GrabCut [24]. We use the supervised version of [23] in our experiments.

Classification over Segmentation Proposals. Inspired by text-based bounding box localization method [23], in this baseline we replace the bounding box proposals in [23] with segmentation proposals (e.g. CPMC [5] and MCG [3]) to output segmentation for the input expression. We use a similar pipeline in this baseline as in the supervised version of [23]. First, visual features are extracted from each proposal and concatenated with the encoded sentence. Then, a classification network is trained on concatenated features to classify a segmentation proposal into either foreground or background. We use 100 top-scoring segmentation proposals from MCG [3], and extract visual features from each proposal by resizing the segmentation proposal regions to 224×224 (i.e. filling pixels outside the proposal region with channel mean and resizing the enclosing bounding box of the proposal) and extracting visual feature from the resized proposal regions with a VGG-16 network pretrained on ILSVRC classification task. The whole network is then trained end-to-end. The main difference between this baseline and our method is that our method performs pixelwise classification through a fully convolutional network, while this baseline requires another proposal method to obtain candidate regions.

Whole Image. As an additional trivial baseline, we also evaluate using the whole image as a segmentation for every expression.

4.2 Evaluation on ReferIt Dataset

We train our model and the baseline methods in Sect. 4.1 on the 10,000 trainval images in the ReferIt dataset [16] (leaving out a small proportion for validation), following the same split as in [15]. In our implementation, we resize and pad all

Table 1. The performance (in %) of our model and baselines on the ReferIt dataset.

Method	*prec*@0.5	*prec*@0.6	*prec*@0.7	*prec*@0.8	*prec*@0.9	overall IoU
whole image	5.07	2.85	1.58	0.81	0.41	15.12
per-word average	10.97	5.94	2.35	0.45	0.00	27.23
per-word intersection	9.58	5.35	2.20	0.43	0.00	26.69
per-word union	10.46	5.65	2.28	0.44	0.00	19.37
SCRC [15] bbox	9.73	4.43	1.51	0.27	0.03	21.72
SCRC [15] grabcut	11.91	7.71	4.33	1.78	0.36	17.84
GroundeR [23] bbox	11.08	6.20	2.74	0.78	0.20	20.50
GroundeR [23] grabcut	14.09	9.62	5.78	2.65	0.62	20.09
MCG classification	12.72	9.88	7.38	4.73	1.88	18.08
Ours (low resolution)	29.54	21.61	13.69	5.94	0.75	45.57
Ours (high resolution)	**34.02**	**26.71**	**19.32**	**11.63**	**3.92**	**48.03**

images and ground-truth segmentation to a fixed size $W \times H$ (where we set $W = H = 512$), keeping their aspect ratio and padding the outside regions with zero, and map the segmentation output back to the original image size to obtain the final segmentation.

In our experiments, we use a two-stage training strategy: we first train a low resolution version of our model, and then fine-tune from it to obtain the final high resolution model (i.e. our full model in Fig. 3). In our low resolution version, we do not add the deconvolution filter in Sect. 3.3, so the model only outputs a $w \times h = 16 \times 16$ coarse response map in Fig. 3. We also downsample the ground-truth label to $w \times h$ and directly train on the coarse response map to match the downsampled label. After training the low resolution model, we construct our final high resolution model by adding a $2s \times 2s$ deconvolution filter with stride $s = 32$, as described in Sect. 3.3, and initialize the filter weights from bilinear interpolation (all other parameters are initialized from low resolution model). The high resolution model is then fine-tuned on the training set using $W \times H$ ground-truth segmentation mask labels. We empirically find this two stage training converges faster than directly training our full model to predict $W \times H$ high resolution segmentation.

We evaluate the performance of our model and the baselines method in Sect. 4.1 on the 10,000 images in the test set. The following two metrics are used for evaluation: the *overall intersection-over-union* (overall IoU) metric and the *precision* metric. The overall IoU is the total intersection area divided by the total union area, where both intersection area and union area are accumulated over all test samples (each test sample is an image and a referential expression). Although the overall IoU metric is the standard metric used in PASCAL VOC segmentation [15], our evaluation is slighly different as we would like to measure how accurate the model can segment the foreground region described by the input expression against the background, and the overall IoU metric favors large regions like sky and ground. So we also evaluate with the precision metric at 5

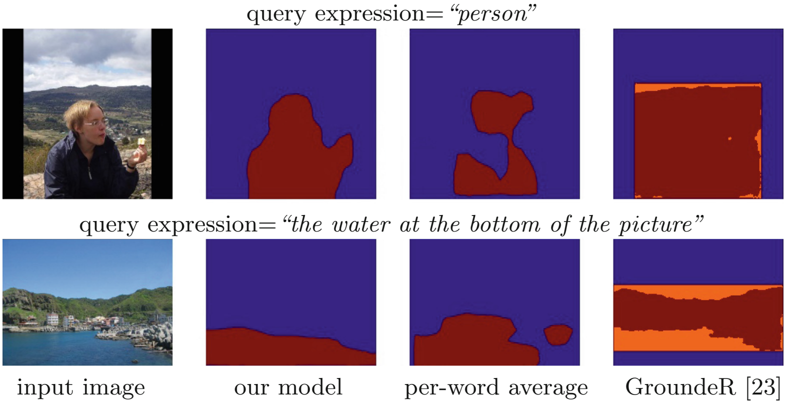

query expression= *"person"*
query expression= *"the water at the bottom of the picture"*
input image
our model
per-word average
GroundeR [23]

Fig. 4. Segmentation examples using our model and baseline methods. For GroundeR [23], the bounding box prediction is in orange and GrabCut segmentation is in red. (Color figure online)

different IoU thresholds from easy to hard: 0.5, 0.6, 0.7, 0.8, 0.9. The precision metric is the percentage of test samples where the IoU between prediction and ground-truth passes the threshold. For example, precision@0.5 is the percentage of expressions where the predicted segmentation overlaps with the ground-truth region by at least 50 % IoU.

Results. The main results for our evaluation are summarized in Table 1. By simply returning the whole image, one already gets 15 % overall IoU. This is partially due to the fact that the ReferIt dataset contains some large regions such as "sky" and "city" and the overall IoU metric put more weights on large regions. However, as expected, the whole image baseline has the lowest precision.

It can be seen from Table 1 that one can get a reasonable overall IoU through per-word segmentation and combining the results from each word. Among the three different ways to combine the per-word results in Sect. 4.1, it works best to average the scores from each word. Using the whole bounding box prediction from SCRC [15] ("SCRC bbox") or GroundeR [23] ("GroundeR bbox") achieves comparable precision to averaging per-word segmentation, while they are worse in terms of overall IoU, and using classification over segmentation proposals from MCG ("MCG classification") leads to slightly higher precision than these two methods. Also, it can be seen that using GrabCut [24] to segment the foreground from bounding boxes ("SCRC grabcut" and "GroundeR grabcut")

Table 2. Average time consumption to segmentation an input (a given image and a natural language expression) using different methods.

Method	per-word	SCRC [15] grabcut	GroundeR [23] grabcut	MCG classification	Ours (high resolution)
time (sec)	0.169	4.319	3.753	9.375	0.325

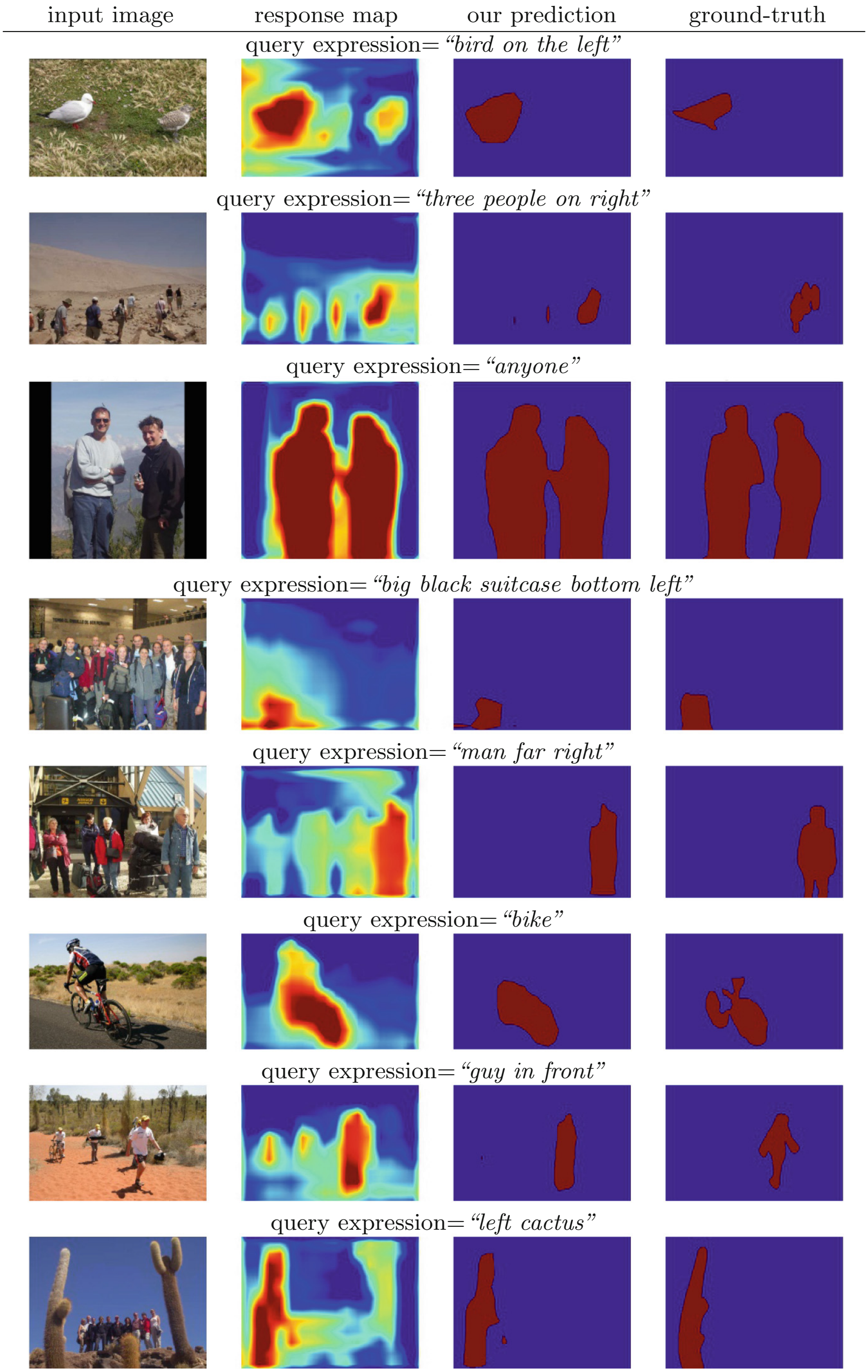

input image
response map
our prediction
ground-truth
query expression= "bird on the left"
query expression= "three people on right"
query expression= "anyone"
query expression= "big black suitcase bottom left"
query expression= "man far right"
query expression= "bike"
query expression= "guy in front"
query expression= "left cactus"

Fig. 5. Segmentation examples on object regions in the ReferIt dataset.

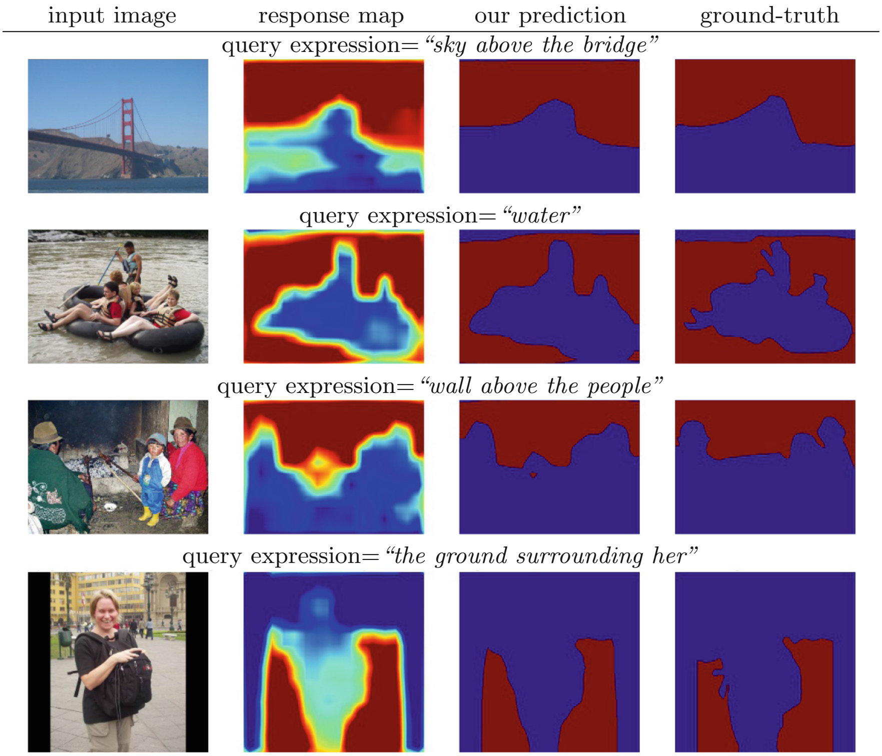

input image
response map
our prediction
ground-truth
query expression= *"sky above the bridge"*
query expression= *"water"*
query expression= *"wall above the people"*
query expression= *"the ground surrounding her"*

Fig. 6. Segmentation examples on stuff regions in the ReferIt dataset.

results in higher precision for both SCRC and GroundeR than using the entire bounding box region. We believe that the precision metric is more reflective for the performance of this task, since in real applications, one would often care more about how often a referential expression is correctly segmented.

Our model outperforms all the baseline methods by a large margin under both precision metric and overall IoU metric. In Table 1, the second last row ("low resolution") corresponds to directly using bilinear upsampling over the coarse response map from our low resolution model, and the last row ("high resolution") shows the performance of our full model. It can be seen that our final model achieves significantly higher precision and overall IoU, compared with the baseline methods. Figure 4 shows some segmentation examples using our model and baseline methods.

The ReferIt dataset contains both object regions and stuff regions. Objects are those entities that have well-defined structures and closed boundaries, such as person, dog and airplane, while stuffs are those entities that do not have a fixed structure, such as sky, river, road and snow. Despite this difference, both object regions and stuff regions can be segmented through our model using the same approach. Figure 5 shows some segmentation examples on object regions

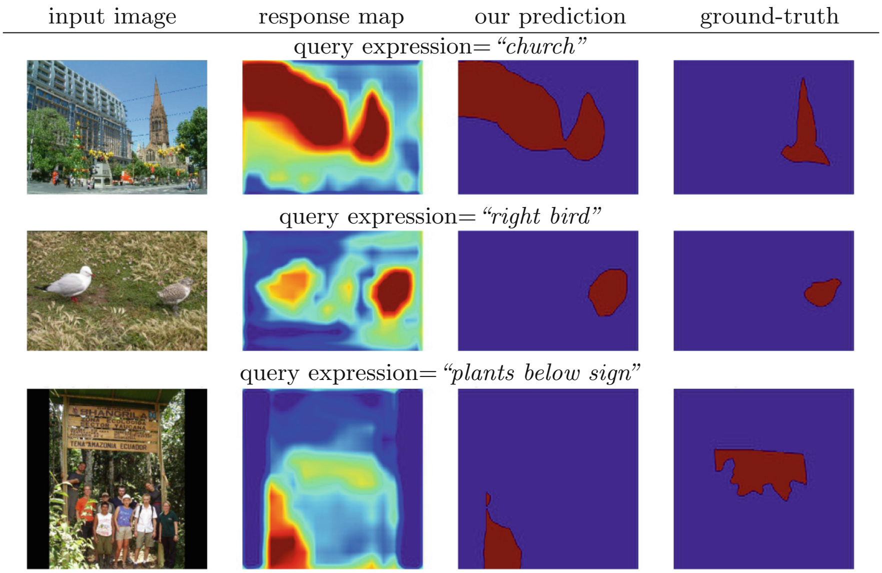

input image
response map
our prediction
ground-truth
query expression= "church"
query expression= "right bird"
query expression= "plants below sign"

Fig. 7. Some failure cases where IoU < 50 % between prediction and ground-truth.

from our model, and Fig. 6 shows examples on stuff regions. It can be seen that our model can predict reasonable segmentation for both object expressions like "bird on the left" and stuff expressions like "sky above the bridge".

Figure 7 shows some failure cases on the ReferIt dataset, where the IoU between prediction and ground-truth segmentation is less than 50 %. In some failure cases (e.g. Fig. 7, middle), our model produces reasonable response maps that cover the target regions of the natural language referential expressions, but fails to precisely segment out the boundary of objects or stuffs.

Speed. We also compare the speed of our method and baseline methods. Table 2 shows the average time consumption for different models to predict a segmentation at test time, on a single machine with NVIDIA Tesla K40 GPU. It can be seen that although our method is slower than the per-word segmentation baseline, it is significantly faster than proposal-based methods such as "SCRC grabcut" or "MCG classification".

5 Conclusion

In this paper, we address the challenging problem of segmenting natural language expressions, to generate a pixelwise segmentation output for the image region described by the referential expression. To solve this problem, we propose an end-to-end trainable recurrent convolutional neural network model to encode the expression into a vector representation, extract a spatial feature map representation from the image, and output pixelwise segmentation based on fully convolutional classifier and upsampling. Our model can efficiently predict segmentation output for referential expressions that describe single or multiple objects or stuffs. Experimental results on a benchmark dataset demonstrate that our model outperforms baseline methods by a large margin.

As the datasets for learning this task directly is limited, we explore in our on-going work [14] how existing large scale vision-only and text-only datasets can be utilized to train our model.

Acknowledgments. This work was supported by DARPA, AFRL, DoD MURI award N000141110688, NSF awards IIS-1427425 and IIS-1212798, and the Berkeley Artificial Intelligence Research (BAIR) Lab. M. Rohrbach was supported by a fellowship within the FITweltweit-Program of the German Academic Exchange Service (DAAD).

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous systems. arXiv:1603.04467 (2016)
2. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE International Conference on Computer Vision (2016)
3. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 328–335 (2014)
4. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7578, pp. 430–443. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33786-4_32
5. Carreira, J., Sminchisescu, C.: CPMC: automatic object segmentation using constrained parametric min-cuts. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1312–1328 (2012)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: Proceedings of the International Conference on Learning Representations (2015)
7. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. In: Syntax, Semantics and Structure in Statistical Translation (2014)
8. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)
9. Escalante, H.J., Hernández, C.A., Gonzalez, J.A., López-López, A., Montes, M., Morales, E.F., Sucar, L.E., Villaseñor, L., Grubinger, M.: The segmented and annotated IAPR TC-12 benchmark. Comput. Vis. Image Underst. **114**(4), 419–428 (2010)

10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC 2012) Results (2012). http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
11. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The IAPR TC-12 benchmark: a new evaluation resource for visual information systems. In: International Workshop OntoImage, pp. 13–23 (2006)
12. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 297–312. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10584-0_20
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Hu, R., Rohrbach, M., Venugopalan, S., Darrell, T.: Utilizing large scale vision and text datasets for image segmentation from referring expressions. arXiv preprint (2016)
15. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
16. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.L.: ReferitGame: referring to objects in photographs of natural scenes. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 787–798 (2014)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
18. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
19. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-RNN). In: Proceedings of the International Conference on Learning Representations (2015)
20. Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J., Murphy, K.P.: Im2Calories: towards an automated mobile vision food diary. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1233–1241 (2015)
21. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)
22. Plummer, B., Wang, L., Cervantes, C., Caicedo, J., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
23. Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 817–834. Springer, Heidelberg (2016)
24. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. (TOG) **23**, 309–314 (2004). ACM
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)

26. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1745–1752. IEEE (2011)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (2015)
28. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
29. Xu, H., Saenko, K.: Ask, attend and answer: exploring question-guided spatial attention for visual question answering. arXiv preprint arXiv:1511.05234 (2015)
30. Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of the International Conference on Machine Learning (ICML) (2015)
31. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE International Conference on Computer Vision (2016)
32. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: Proceedings of the International Conference on Learning Representations (2016)
33. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
34. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 391–405. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10602-1_26