See-Through-Text Grouping for Referring Image Segmentation

Ding-Jie Chen[1], Songhao Jia[2], Yi-Chen Lo[2], Hwann-Tzong Chen[2], and Tyng-Luh Liu[1]
[1]Institute of Information Science, Academia Sinica, Taiwan
[2]Department of Computer Science, National Tsing Hua University, Taiwan

Abstract

Motivated by the conventional grouping techniques to image segmentation, we develop their DNN counterpart to tackle the referring variant. The proposed method is driven by a convolutional-recurrent neural network (ConvRNN) that iteratively carries out top-down processing of bottom-up segmentation cues. Given a natural language referring expression, our method learns to predict its relevance to each pixel and derives a See-through-Text Embedding Pixelwise (STEP) heatmap, which reveals segmentation cues of pixel level via the learned visual-textual co-embedding. The ConvRNN performs a top-down approximation by converting the STEP heatmap into a refined one, whereas the improvement is expected from training the network with a classification loss from the ground truth. With the refined heatmap, we update the textual representation of the referring expression by re-evaluating its attention distribution and then compute a new STEP heatmap as the next input to the ConvRNN. Boosting by such collaborative learning, the framework can progressively and simultaneously yield the desired referring segmentation and reasonable attention distribution over the referring sentence. Our method is general and does not rely on, say, the outcomes of object detection from other DNN models, while achieving state-of-the-art performance in all of the four datasets in the experiments.

1. Introduction

The rapid development of deep neural networks (DNNs) and the availability of large-scale image/video datasets have prompted significant research progress on segmentation problems such as semantic segmentation [4, 34], instance segmentation [9, 21], and interactive segmentation [42]. However, while high accuracy on semantic labeling for these tasks can be achieved by state-of-the-art methods, a notable shortcoming of such applications is that the label semantics are usually predefined by restricted object classes.

In practical scenarios, referring to an object or objects of interest through natural language expressions instead of the predefined class labels should be more attractive to users. Since a natural language expression may comprise attributes, actions, spatial relationship, and interaction for characterizing the visual entities, the abundant expression thus provides flexibility. The field of natural language processing (NLP) has developed several useful language models for extracting language features [2, 30, 31, 32]. Benefited from such convenience, language-based visual understanding also gains much attention and has been applied to the tasks of visual question answering [1, 11, 27, 47], referring object localization [14, 20, 44], image captioning [15, 38], and referring image segmentation [12, 13, 19, 23, 29].

We aim to address the problem of *referring image segmentation*, in which a natural language referring expression is provided to guide pixel-level image segmentation. Referring image segmentation can be treated as richer-class semantic segmentation. This kind of technique serves as a versatile human-machine interaction mechanism for interactive image segmentation. With the interaction mechanism, users can provide natural language sentences as descriptions via typing or speaking for guiding the machine to select the region of interest [17]. However, the flexibility of natural language expressions for referring rich object classes also means that referring image segmentation is challenging—a region (or regions) being referred to can be anywhere the natural language expressions are able to describe.

Previous results [12, 13, 19, 23, 29] on referring image segmentation often follow a "concatenation-convolution" procedure. These methods mainly concatenate visual and language features extracted from the given image and referring expression, respectively. The procedure is followed by applying convolution operations to the concatenated features. Such techniques essentially seek an optimal weighted average, along the channel dimension of the concatenated features, for yielding the segmentation. They do not jointly consider how visual features of the pixels in a referred region correlate with the natural language expression, but still achieve reasonable performance with powerful DNNs.

Prior to the widespread use of deep learning, methods to address image segmentation can be categorized into the bottom-up approach, the top-down approach and the integration of the two, *e.g.*, [3]. A bottom-up technique focuses on grouping pixels into coherent regions, while a top-down method explores prior information such as object representation to accomplish the task. Our method for referring image segmentation is a DNN approach that realizes the fusion of the bottom-up and the top-down viewpoints.

Intuitively, given a training dataset for referring image segmentation, it is feasible to learn a compatibility measure such that pixels within a referred region yield high compatibility scores to the referring expression, while the opposite have low compatibility scores. As illustrated in Figure 1, the proposed model includes a component to collect bottom-up segmentation cues of pixel level by leveraging the compatibility measure from the visual-textual co-embedding, which is named as See-through-Text Embedding Pixelwise (STEP). On the other hand, the ground-truth information in computing the classification loss can teach the model to refine each STEP heatmap into a top-down approximation. Observe that a proper textual representation would benefit the prediction of a segmentation heatmap, and analogously a revealing segmentation heatmap can help revise the textual representation. Therefore, we design a ConvRNN model [36] to iterate the steps of combining bottom-up and top-down information so that the bilateral improvements just described can be further explored to yield satisfactory results of referring image segmentation. We characterize the main advantages of our method as follows:

- Different from the "concatenation-convolution" procedure, our method explicitly learns a visual-textual co-embedding, termed as See-through-Text Embedding Pixelwise (STEP) to align the two modalities.
- The proposed STEP yields a compatibility measure to the given referring natural language expression, and in turn enables associating relevant pixels into coherent regions. The grouping principle resembles the conventional bottom-up approach for image segmentation.
- The more precise the textual representation is, the better the result of referring segmentation is. The design principle of the architecture includes a ConvRNN and proper weighting/attention schemes to ensure the iterative fusion process alternately improves the two aspects and boosts the final segmentation result.
- The proposed method is a stand-alone DNN approach whose architecture elegantly realizes the fusion of bottom-up and top-down reasonings for the segmentation task. Furthermore, it achieves state-of-the-art performance in all our experiments without leveraging additional training data as well as other DNN models.

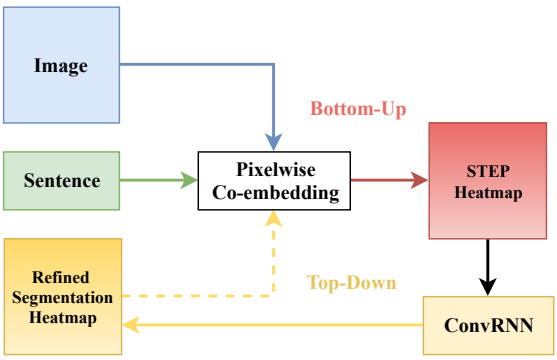

Figure 1. See-through-Text Grouping: At each time step, bottom-up grouping of pixelwise visual-textual co-embedding yields a See-through-Text Embedding Pixelwise (STEP) heatmap. Then the top-down process by ConvRNN converts the input STEP into a refined one, which is used to update the textual representation of the referring expression for the ensuing time step of ConvRNN.

2. Related Work

We give a concise overview about recent research efforts relevant to the problem of referring image segmentation.

2.1. Semantic Segmentation and Embeddings

Many recent semantic segmentation models are built on the idea of Fully Convolutional Network (FCN) [26]. FCN replaces the fully connected layers with convolutional layers and uses skip-connection for generating dense pixel-level labeling. DeepLab [4] presents atrous convolution to preserve the spatial resolution for keeping the receptive field during convolution. In the visual-textual co-embedding step of our approach, we use the atrous convolution to make each pixel combine multi-context visual representations with various sizes of the receptive field. DeViSE [7] is one of the representative approaches to visual-semantic embedding. Later, Wang *et al*. consider the structure-preserving constraints in learning a joint embedding for image-to-text and text-to-image retrieval [39]. Rohrbach *et al*. propose to ground a phrase by soft attentions over bounding box proposals [33]. Our method also computes attention, but ours is at pixel-level instead of box-level.

2.2. Referring Expression Comprehension

Integrating computer vision and natural language is an active research area. Instead of using an object detector, the methods for object localization [14] and object tracking [20] are able to localize the object regions that are specified by natural language expressions. In [28, 45], both methods simultaneously localize the object specified by the language and generate the description per object. Liu *et al*. [25] present an attribute learning and embedding method to show the usefulness of visual attributes in referring expression comprehension. Neighbourhood Watch [40] uses self-

attention to decompose expression into three components. It achieves state-of-the-art performance on referring expression comprehension. The task of referring image segmentation is also related to visual question answering (VQA) [1, 11, 27, 47], which usually fuses multi-modal features in their models. Methods for VQA usually use a Recurrent Neural Network (RNN) to encode or generate sequences. We also use regular RNN to encode sentence. However, our approach additionally introduces a way to apply a convolutional RNN for fusing multiple heatmaps. Besides, the ideas of joint embedding and learning the relatedness of different representations have been adopted in VQA [41]. However, instead of using average pooling to transform each image region into a vector for the subsequent embedding, we densely embed the per-pixel visual representation into a common space. The proposed See-through-Text Embedding Pixelwise (STEP) provides contextual information for subsequent top-down grouping.

2.3. Referring Image Segmentation

This section reviews several state-of-the-art methods on the task of referring image segmentation, and then indicates the differences from our approach.

Hu *et al*. [12] first propose to solve the task of referring image segmentation. They present the "concatenation-convolution" model which concatenates the language and image features and then performs convolution on the concatenated features to generate the segmentation. The language representation is obtained via Long Short-Term Memory networks (LSTM) [10]; the visual representation is obtained via the VGG-16 network [37]. They then further improve their model using extra vision-only and image-only training data [13]. Liu *et al*. [23] suggest using the sequential nature of language. Their method concatenates multi-modal features during processing every new word. These concatenated features are then fused via multi-modal LSTM (mLSTM) as a joint representation for feeding to the final convolution. They also use LSTM to encode the language representation but use DeepLab ResNet-101 [4] to encode the visual representation. Li *et al*. [19] improve the model of [12] by considering the multi-scale semantics possessed in the visual representation encoding step. Their feature fusion step employs convolutional LSTM [36]. Margffoy-Tuay *et al*. [29] concatenate richer visual and language features, which comprise word embeddings, per-word hidden state (via simple recurrent units [18]), response to dynamic filters [20], and visual features (via DPN92 [5]).

MAttNet [44] achieves state-of-the-art performance on both bounding-box-level and pixel-level comprehension tasks. Neural Module Tree network (NMTree) [24] also shows similar performance as MAttNet. However, both of them rely on an additional detector, *e.g*. Faster-RCNN, for pre-processing to extract ROI features, and then need an additional pixel annotator, *e.g*. Mask-RCNN, for post-processing to obtain segmentation.

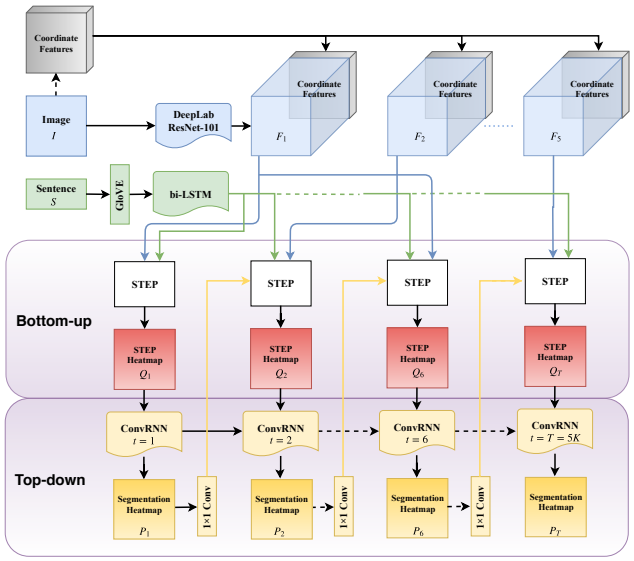

Figure 2. The proposed DNN architecture to referring image segmentation, with K-fold see-through-text grouping. The green lines indicate the language representation flow. The blue lines indicate the visual representation flow. The five sets of feature maps (from five conv layers) implies ConvRNN has $5 \times K$ time steps.

To sum up: Most of the previous methods [12, 13, 19, 23, 29] of referring image segmentation follow the "concatenation-convolution" notion as [12, 13]. With the concatenated multi-modal features, these methods may focus on the sequential nature of language [23, 29] or richer visual representations [29], with the consideration of the multi-resolution visual features [19, 29] for raising the segmentation accuracy. Finally, convolution layers are used to fuse the concatenated multi-modal features for generating the final segmentation. In contrast, our approach explicitly learns a co-embedding for measuring the compatibility of multi-modal features, naturally leading to bottom-up groupings for image segmentation. We further use a ConvRNN to enable the top-down integration of the bottom-up segmentation cues. Our motivations and the network architecture significantly differ from the aforementioned techniques.

3. Our Method

Figure 2 shows an overview of our approach to referring image segmentation. Given an input image I and a natural language expression S as the referring hint, the task is to localize the foreground regions specified by S. The proposed see-through-text grouping is driven by a ConvRNN. At each time step t, it first decides the visual representation of I from the respective feature maps of the adopted feature extraction model, and encodes S into a proper textual representation, which is computed by taking account

of the segmentation prediction from the previous time step. The bottom-up grouping jointly embeds the two representations into a common feature space for pixelwise measuring their multi-modal compatibility and yields the STEP heatmap Q_t, indicating the foreground probability of each pixel. Taking Q_t as the input, the ConvRNN refines it into P_t, the probability heatmap for the referring segmentation at time t. Upon completing the iterative process, our method arrives at the final segmentation result, say, P_T.

3.1. Visual Representation

Following [23], we use DeepLab ResNet-101v2 [4] pre-trained on Pascal VOC [6] to generate the visual representation. All input images are re-sized to $W \times H$ with zero-padding. Within the model, we focus on the five convolutional layers and use the notations $F_\ell, \ell \in \{1, 2, 3, 4, 5\}$ to denote the feature maps generated from the corresponding convolutional layer ℓ. To enrich the visual representation, we also encode the spatial relationships by directly concatenating an 8-dimensional spatial coordinate representation [14, 23] to each F_ℓ. Note that the ConvRNN is performed from $t = 1, \ldots, T$ where $T = 5 \times K$ indicates a K-fold implementation of our method. At time step t, the visual representation is determined by F_{ℓ_t} with $\ell_t = t \bmod 5$.

Specifically, we set the input size $W = H = 320$ and hence the resolution is 80×80 for each feature map from F_1 and F_2, and 40×40 for each from F_3, F_4, and F_5. Further, the number of channels of F_1, F_2, F_3, F_4, and F_5 are 64, 256, 512, 1024, and 2048, respectively. The complete visual representations (for each pixel) with respect to F_1, F_2, F_3, F_4, and F_5 are of sizes 72, 264, 520, 1032, and 2056, after adding the 8-dimensional spatial coordinate representation.

3.2. Textual Representation

We use $S = \{w_1, w_2, \ldots, w_n\}$ to denote the given natural language expression. To derive the textual representation at time t, denoted as $\mathbf{s}_t$, we consider the pre-trained GloVe model [32] and encode each word $w_i \in S$ into a 300-D GloVe word embedding $\mathbf{w}_i \in \mathbb{R}^{300}$. The input sentence S is then expressed by a concatenation of each GloVe word embedding. Notice that the textual representation of S will be gradually improved if the referred region in I could become more evident. To this end, we concatenate the n word embeddings of S and feed them into a one-layer bi-directional LSTM (biLSTM). Let $\mathbf{h}_j$ be the hidden-state output after running the biLSTM through the first j words of S. Also let $\mathbf{v}_i$ be the visual feature vector of pixel i in I. The dimension of $\mathbf{v}_i$ varies with respect to F_{ℓ_t}. However we respectively attach 1×1 convolution to each hidden state of biLSTM and each visual feature $\mathbf{v}_i$ such that $\mathbf{h}_j \mapsto \tilde{\mathbf{h}}_j \in \mathbb{R}^{400}$ and $\mathbf{v}_i \mapsto \tilde{\mathbf{v}}_i \in \mathbb{R}^{400}$. Then the textual representation of S at time t of running the pixelwise co-embedding is

$$\mathbf{s}_t = \sum_{i \in I} \pi\{P_{t-1}(i)\} \times \sum_{j=1}^{n} \pi\{\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{h}}_j \rangle\}\, \mathbf{h}_j \qquad (1)$$

where $\pi\{\cdot\}$ denotes the $\mathrm{Softmax}$ function and $P_{t-1}(i)$ is the probability of pixel i being the referring foreground, according to the segmentation heatmap from the previous step. The representation $\mathbf{s}_t$ is visual-attended and its goodness is linked to the predicted segmentation map P_{t-1}.

The GloVe model in our implementation is pre-trained on Common Crawl in $840B$ tokens. Following [19, 23], we keep only the first 20 words per sentence. In addition, we set the cell size of the biLSTM module as 1000.

3.3. See-through-Text Embedding

To compute STEP for each time step t of ConvRNN, we first denote the per-pixel visual representation as $\mathbf{v}$. Consider now an arbitrary pair of visual-textual representations $\mathbf{v}$ and $\mathbf{s}$. (For simplicity, we omit the subscript t in $\mathbf{s}_t$ as the following discussion is valid for all t.) We learn the respective joint embedding functions ϕ and ψ of each modality. The two mappings $\phi(\mathbf{v})$ and $\psi(\mathbf{s})$ are expected to embed related $(\mathbf{v}, \mathbf{s})$ pairs into nearby neighborhoods in the space of visual-textual co-embedding. Hence, the task of pixel association can be reduced to predicting the relationship between these two different modal representations via the learned corresponding embedding functions.

While the sequential nature of language data can be well represented via a recurrent neural network, the spatial nature of visual data can be distilled by enlarging the field of view of filters to incorporate more context information. In our formulation, besides using biLSTM to encode the language hint, we consider atrous convolution [4] to gain context (field-of-view) information. In particular, all feature maps from F_ℓ, $\ell \in \{1, 2, 3, 4, 5\}$ are generated with atrous convolutions from kernel size of 3×3 in rate $r = 3$.

Therefore, given the pair representations $\mathbf{v}$ and $\mathbf{s}$, STEP generates their embeddings $\phi(\mathbf{v})$ and $\psi(\mathbf{s})$ via a normalized single layer fully-connected (fc) network by

$$\phi(\mathbf{v}) = \mathrm{NL}_2(\tanh(W_v \cdot \mathbf{v} + \mathbf{b}_v))\,, \qquad (2)$$

$$\psi(\mathbf{s}) = \mathrm{NL}_2(\tanh(W_s \cdot \mathbf{s} + \mathbf{b}_s))\,, \qquad (3)$$

where W and $\mathbf{b}$ are weights and bias of the fc networks and $\mathrm{NL}_2(\cdot)$ denotes the $L2$-normalization. The output dimensions of both networks in (2) and (3) are set as 1000.

3.4. Bottom-up STEP Heatmaps

We can now readily use the cosine similarity to measure the compatibility of each visual-textual pair $(\mathbf{v}, \mathbf{s})$ in the space of co-embedding in that $L2$-normalization has been applied to both embedding features by ϕ in (2) and ψ in

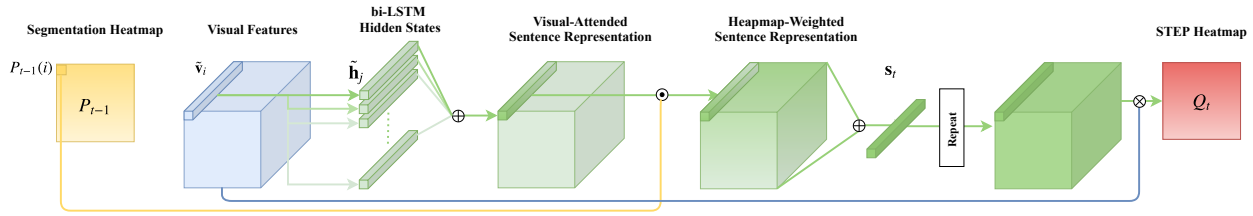

Figure 3. Our method first computes per-pixel textual representations and then yields their weighted combination by referencing the segmentation hint form P_{t-1}. Once the textual representation $\mathbf{s}_t$ is updated by (1), the STEP heatmap Q_t can be obtained from (4).

(3). That is, the STEP compatibility at pixel i with visual features $\mathbf{v}_i$ can be obtained via inner product:

$$Q(i) = \max\{0,\ \langle \phi(\mathbf{v}_i), \psi(\mathbf{s}) \rangle\}\,. \tag{4}$$

The core idea of the bottom-up grouping process is assuming that the per-pixel compatibility of a visual-textual pair measured in the space of co-embedding is proportional to the correlation of the pair. Namely, a visual-textual pair $(\mathbf{v}_i, \mathbf{s})$ of high compatibility means the visual representation at pixel i has a high probability of being referred by the natural language representation $\mathbf{s}$. We illustrate the key operations for predicting the STEP heatmap in Figure 3.

3.5. Top-down Heatmap Refinement

The STEP heatmap Q_t derived at time step t results from a bottom-up grouping process, where pixel association is achieved by a compatibility measure to a given textual representation $\mathbf{s}_t$. Such a grouping process heavily relies on *local correlations* and thus lacks a global view of the desired referring segmentation. Motivated by the success of combining bottom-up and top-down processing for image segmentation, we learn a top-down process driven by the already-mentioned ConvRNN to refine Q_t with the guidance from the ground truth of referring segmentation.

In our method we choose to implement the top-down heatmap refinement with the ConvRNN as in our pilot testing, heatmap refinement with the convolutional gated recurrent units yields satisfactory results. Specifically, we adopt the convolutional GRU as the base model in the experiments. The convolutional GRU with input $\{x_t\}$ is represented with the following equations.

$$f_t = \sigma(R^f * h_{t-1} + W^f * x_t + b^f)\,, \tag{5}$$
$$z_t = \sigma(R^z * h_{t-1} + W^z * x_t + b^z)\,, \tag{6}$$
$$\hat{h}_t = \tanh(R^h * (f_t \odot h_{t-1}) + W^h * x_t + b^h)\,, \tag{7}$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t\,, \tag{8}$$

where f_t, z_t, and h_t are reset gate values, update gate values, and hidden activations at frame t, respectively. The weights of the input and recurrent hidden units are W^* and R^*. The biases are b. σ denotes the sigmoid function and $\odot$ denotes the element-wise multiplication product. The GRU combines the input and forget gates into an update gate z_t for balancing the previous activation h_{t1} and the update activation $\hat{h}_t$. The reset gate f_t decides whether or not to forget the previous activation.

The final hidden state h_T from the ConvGRU comprises the multi-resolution information for predicting the referring foreground probability. We then use a 1×1 convolutional layer to obtain the final probability heatmap

$$P_T = \sigma(W^P * h_T + b^P)\,. \tag{9}$$

Details The weights of the convolutional GRU are of size $h \times w \times c \times f$, where h, w, c, and f respectively denote the kernel's height, width, number of input channels, and number of filters. In all our experiments, we set $h = w = c = 3$ and $f = 32$. Also note that, as illustrated in Figure 2, the K-fold implementation of our method implies that there are totally $5 \times K$ STEP heatmaps $\{Q_t\}$ as the input to the convolutional GRU to yield the final referring segmentation.

3.6. Training

The complete DNN model elegantly connects the two coupled bottom-up and top-down processes. To make the network end-to-end trainable, we use bi-linear interpolation to upsample the probability heatmap derived in (9) by

$$P_T \xrightarrow{\text{upsample}} P \in \mathbb{R}^{W \times H}. \tag{10}$$

It follows that given the binary ground-truth mask G of an input image I for referring image segmentation, we define the binary cross-entropy loss function of our model as

$$L = \frac{-1}{HW} \sum_i \{G \log P + (1 - G) \log(1 - P)\}(i)\,. \tag{11}$$

With (11), our network is learned using Adam optimizer and stop training after $700K$ iterations. The weight decay and the initial learning rate are 0.0005 and 0.00025. We use a polynomial decay with power of 0.9.

4. Experiments

We evaluate our model in two metrics on four datasets as [19, 23, 29]. The first experiment compares multiple variants of the proposed model. We then evaluate the segmentation accuracy of our model against state-of-the-arts.

VARIANTS	UNC		
	val	testA	testB
Ours (1-fold)	56.68	58.70	55.39
no GloVe-biLSTM	55.11	56.87	53.55
no LTR	54.40	56.09	52.59
1x1-conv	48.71	48.35	48.43
convGRU	55.97	58.16	54.73
concate-conv	48.08	49.07	47.35
ini-LTR-uniform	56.74	58.53	55.30
reverse STEP	57.01	58.50	55.25

Table 1. Comparison of the UNC dataset using the first metric, *i.e.*, mIoU, against downgraded versions, each of which results from considering the listed component rather than our implementation.

ReferItGame (ReferIt) [16]: ReferIt comprises 130,525 language expressions referring to 96,654 object regions in 19,894 natural images. The segmentation targets in this dataset consists of object and stuff (*e.g.*, water and sky). The language expressions in ReferIt are usually shorter and more concise [23]. We use the same split as [12].

UNC & UNC+ [45]: Both UNC and UNC+ are collected from MS COCO images. UNC contains 142,209 language expressions for 50,000 objects in 19,994 images, and UNC+ consists of 141,564 expressions referring to 49,856 objects in 19,992 images. The dataset UNC has no restrictions on the referring expressions, while the dataset UNC+ does not allow location-describing words in the expressions. Namely, an expression annotator has to describe the object purely by its appearance. We use the same split as [45].

Google-Ref (GRef) [28]: GRef contains 104,560 referring expressions for 54,822 objects in 26,711 images selected from MS COCO dataset [22]. The images contain 2 to 4 objects of the same type. The language expressions are longer and with richer descriptions [23]. We use the same data split as [28].

Metrics: As [8, 12, 23], the first metric is mean intersection-over-union metric (*mIoU*), which collects total interaction regions over the total union regions of all the test images. The second metric is the precision evaluated from 0.5 to 0.9, *i.e.*, Prec@X, where $X \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

4.1. Ablation Study

This experiment compares several variants of the proposed model for assessing the relative importance of each configuration, and the results on the UNC dataset are shown in Table 1 and Table 2.

METHODS	Prec@0.5	Prec@0.6	Prec@0.7	Prec@0.8	Prec@0.9	mIoU
RMI [23]	41.27	29.71	18.41	7.37	0.76	44.33
RMI+DCRF [23]	42.99	33.24	22.75	12.11	2.23	45.18
RRN [19]	60.19	50.19	38.32	23.87	5.66	54.26
RRN+DCRF [19]	61.66	52.50	42.40	28.13	8.51	55.33
Ours (1-fold)	66.03	55.91	44.35	27.65	7.43	56.68
Ours (2-fold)	67.72	58.70	47.03	30.86	8.13	57.23
Ours (4-fold)	**70.15**	**63.37**	**53.15**	**36.53**	**10.45**	**59.13**

Table 2. Comparison based on the second metric, *i.e.*, Prec@X, on the UNC val split.

In Table 1, each row shows one of the following settings respectively: our full method in 1-fold (Ours), without using the pre-trained GloVe embedding and the bi-directional LSTM (no GloVe-biLSTM), without learning the textual representation with (1) (no LTR[1]), replacing convLSTM with convGRU (convGRU), replacing ConvRNN with 1x1 convolution (1x1-conv), replacing STEP with concatenation-convolution procedure (concate-conv), learning the first textual representation with uniform heatmap (ini-LTR-uniform), generating the STEP heatmaps from high-level visual feature to low-level visual feature (reverse STEP).

For the case of learning textual representation, without using the pre-trained GloVe embedding and the bi-directional LSTM (no GloVe-biLSTM), the performance decreases from 1.5% to 1.8%. Without using the proposed textual representation learning (no TRL), the accuracy decreases more from 1.8% to 2.7%. According to the comparison between the two textual-representation-related factors, the gain of using our textual representation learning approach, *i.e.* (1), is clear. This experiment demonstrates that using the hidden state of ConvRNN to guide the learning of textual representation is beneficial to the segmentation task.

Considering the strategies for top-down heatmap refinement, using the 1x1 convolution to integrate the STEP heatmaps causes obvious performance drop from 6.9% to 10.4%. Replacing the convolutional LSTM with a convolutional GRU slightly decreases the accuracy from 0.5% to 0.7%. This comparison shows that integrating multiple heatmaps via ConvRNN is beneficial to the segmentation performance. For training our model in multiple folds, we use the ConvGRU to save more computational cost.

We also try to learn the first textual representation with a uniform heatmap instead of directly using the final hidden state from biLSTM (ini-LTR-uniform) and try to reverse the order to generate the STEP heatmap from high-level to low-level visual features (reverse STEP). These two results show that our model is not sensitive to these two factors.

Besides, we additionally replace the embedding scheme in STEP procedure with a concatenation-convolution

[1]We merely use the final hidden-state of biLSTM in STEP procedure.

TYPE	METHOD	ReferIt	UNC			UNC+			GRef
		test	val	testA	testB	val	testA	testB	val
RIS	LSTM-CNN [12]	48.03	-	-	-	-	-	-	28.14
	LSTM-CNN+ [13]	49.91	-	-	-	-	-	-	34.06
	RMI+DCRF [23]	58.73	45.18	45.69	45.57	29.86	30.48	29.50	34.52
	RRN+DCRF [19]	63.63	55.33	57.26	53.95	39.75	42.15	36.11	36.45
	DMN [29]	52.81	49.78	54.83	45.13	38.88	44.22	32.29	36.76
	KWAN [35]	59.19	-	-	-	-	-	-	36.92
	Ours (5-fold)	**64.13**	**60.04**	**63.46**	**57.97**	**48.19**	52.33	**40.41**	**46.40**
RIL	MAttNet [44]	-	56.51	62.37	51.70	46.67	**52.39**	40.08	n/a
	SLR [46]	not evaluated on pixelwise segmentation							

Table 3. Experimental results of mIoU metric on four datasets. "-" indicates no available results. "n/a" denotes the method does not use the same split as the RIS ones. Note that our method does not rely on either detectors as pre-processing or segmentation for post-processing.

scheme. Precisely, we concatenate the visual representation and textual representation followed by an MLP for obtaining a one-channel image. The replacement shows the substantial degradation and hence indicates that the one-channel image obtained by the concatenation-convolution scheme may not directly be used for guiding the textual representation learning.

Table 2 compares our model against two referring image segmentation methods. This comparison demonstrates that our basic model in one round has better performance than other methods before using dense CRF (DCRF) as post-processing. Instead of using disposable post-processing, our method shows the ability to increase the performance in multiple folds. The significant gain shows that our model performs well for the referring image segmentation task.

The ablation study justifies that our approach measures the compatibility of multi-modal features via a co-embedding rather than a concatenation-convolution scheme, naturally leading to bottom-up groupings for image segmentation. Further, applying ConvRNN is able to boost the performance for the top-down integration of the bottom-up segmentation cues.

4.2. Comparison with the State-of-the-arts

This section compares the performance in mIoU of our model against state-of-the-art techniques [12, 13, 19, 23, 29, 35] on the task of referring image segmentation (RIS) (*input:* a sentence plus an entire image, *output:* image segmentation). As mentioned earlier, there exists a related task of referring image localization (RIL) [44, 46] (*input:* image crops, *e.g.* RoIs/proposals, *output:* bounding boxes). Since MAttNet [44] transforms their box-result to segmentation as an extension, this experiment includes them for comparison. However, please notice that, as the RIL models often rely on other techniques for pre-processing and post-processing, it may not be appropriate to directly compare their performance with the RIS methods. Table 3 summaries the results of the comparisons.

The performance of our method is significantly better than all those for the task of referring image segmentation, and also outperforms MAttNet in five of the six testing splits, despite that MAttNet relies on Mask R-CNN for proposal generation and semantic segmentation. Note that Mask R-CNN is trained with noticeable more COCO images for detection and segmentation (RoI generator and Mask branch). In addition, MAttNet needs to train an attribute predictor with extra training data of attribute words. In comparison, our method and most of the RIS approaches train their model only with the split of each dataset, *i.e.*, without accessing extra training data as well as the additional attribute words. The results indicate that our method achieves satisfactory performance in learning the relationship between the specific image region and the natural language expression referring to it.

4.3. Qualitative Results

Figure 4 shows the learned word attention and the learned segmentation heatmaps of ConvRNN. The similarity [43] reflects the word attention of our model in each time-step (row). The results show that our model is able to pay attention to different words per time-step. Further, the learned segmentation heatmaps show that our model can progressively adjust the hidden states for sketching the preferred region.

In Figure 5, we provide other qualitative results of referring image segmentation task by our method. This figure shows that a referring image segmentation algorithm can be guided by different query expression for segmenting different regions of interest. (More experimental results are provided in the supplementary material.)

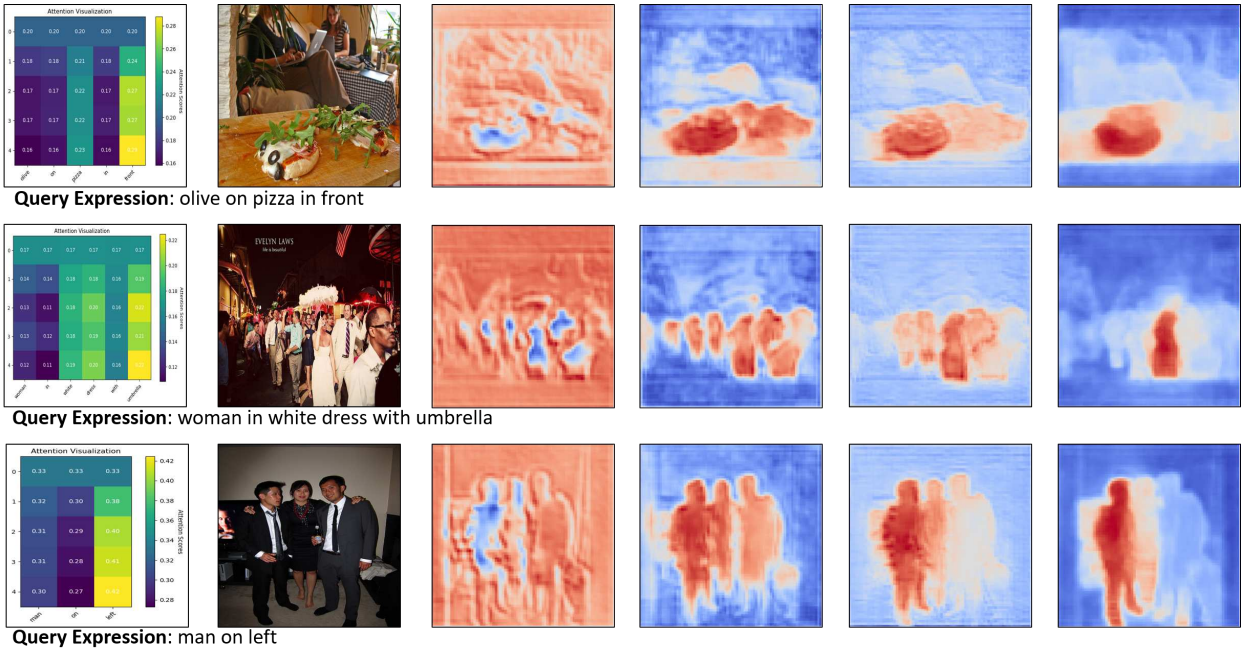

Figure 4. Qualitative segmentation results (ini-LTR-uniform). From left to right per row: the word attention, the input image, and the four progressively learned segmentation heatmaps. Note that the segmentation heatmaps $\{P_t\}$ are generated by 1x1 convolution. In the plot of word attention, each row corresponds to one time step of ConvRNN, each column corresponds to one word as annotated below, and the value of each block is defined as the similarity between the learned textual representation in that time step and the textual representation (hidden state) of that word. We show only five time steps for the sake of clarity. The right images in four columns denote the segmentation heatmaps learned in different time steps of ConvRNN.

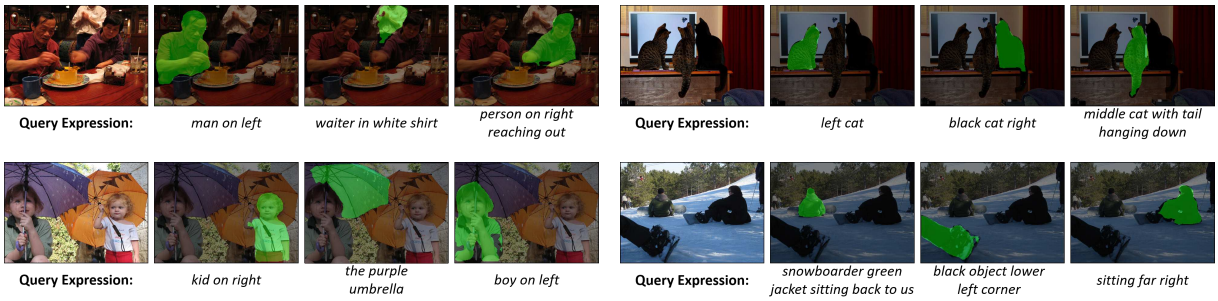

Figure 5. Qualitative results of referring image segmentation by our method.

5. Conclusion

At the core of our method is a new DNN architecture comprising two coupled modules to better solve the referring image segmentation. The first module tackles the problem from the viewpoint of bottom-up grouping, leading to the formulation of the proposed See-through-Text Embedding Pixelwise (STEP). Benefited from the powerful feature learning of a DNN, the per-pixel representation indeed includes informative context from the underlying receptive field. As a result, the resulting STEP heatmap predictions include meaningful and constructive cues to the target segmentation, owing to explicitly learning a compatibility measure via the visual-textual co-embedding. The advantage is further justified in our promising experimental results. The second module of our approach is the use of ConvRNN as the top-down driving mechanism to refine the generated STEP heatmaps. A notable novelty in our design is that the reliability of the input to ConvRNN can be enhanced by referencing the outcome of the previous step. We have carried out a detailed ablation study to verify that all components used in our model positively contribute to the satisfactory performance. Our future work would focus on generalizing the proposed model to referring video segmentation.

Acknowledgement. This work was supported in part by the MOST, Taiwan under Grants 108-2634-F-001-007 and 106-2221-E-007-080-MY3. We are also grateful to the *National Center for High-performance Computing* for providing computational resources and facilities.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *ICCV*, pages 2425–2433, 2015. 1, 3

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017. 1

[3] Eran Borenstein and Shimon Ullman. Combined top-down/bottom-up segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 30(12):2109–2125, 2008. 2

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 1, 2, 3, 4

[5] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *NIPS*, pages 4470–4478, 2017. 3

[6] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 4

[7] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS.*, pages 2121–2129, 2013. 2

[8] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees G. M. Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018. 6

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 1

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 3

[11] Hexiang Hu, Wei-Lun Chao, and Fei Sha. Learning answer embeddings for visual question answering. In *CVPR*, 2018. 1, 3

[12] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, pages 108–124, 2016. 1, 3, 6, 7

[13] Ronghang Hu, Marcus Rohrbach, Subhashini Venugopalan, and Trevor Darrell. Utilizing large scale vision and text datasets for image segmentation from referring expressions. *CoRR*, abs/1608.08305, 2016. 1, 3, 7

[14] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, pages 4555–4564, 2016. 1, 2, 4

[15] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015. 1

[16] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 6

[17] Gierad Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. Pixeltone: a multimodal interface for image editing. In *CHI*, pages 2185–2194, 2013. 1

[18] Tao Lei, Yu Zhang, and Yoav Artzi. Training rnns as fast as cnns. *CoRR*, abs/1709.02755, 2017. 3

[19] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, 2018. 1, 3, 4, 5, 6, 7

[20] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees G. M. Snoek, and Arnold W. M. Smeulders. Tracking by natural language specification. In *CVPR*, pages 7350–7358, 2017. 1, 2, 3

[21] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2978–2991, 2018. 1

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 6

[23] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan L. Yuille. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, pages 1280–1289, 2017. 1, 3, 4, 5, 6, 7

[24] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. Explainability by parsing: Neural module tree networks for natural language visual grounding. *CoRR*, abs/1812.03299, 2018. 3

[25] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *ICCV*, pages 4866–4874, 2017. 2

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2

[27] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, pages 289–297, 2016. 1, 3

[28] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 2, 6

[29] Edgar Margffoy-Tuay, Juan C. Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, pages 656–672, 2018. 1, 3, 5, 7

[30] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010. 1

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013. 1

[32] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 1, 4

[33] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, pages 817–834, 2016. 2

[34] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017. 1

[35] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *ECCV*, 2018. 7

[36] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015. 2, 3

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3

[38] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015. 1

[39] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013, 2016. 2

[40] 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. *CoRR*, abs/1812.04794, 2018. 2

[41] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017. 3

[42] Ning Xu, Brian L. Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016. 1

[43] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL HLT*, pages 1480–1489, 2016. 7

[44] 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 *CVPR*, 2018. 1, 3, 7

[45] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 2, 6

[46] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 7

[47] Yuke Zhu, Oliver Groth, Michael S. Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, pages 4995–5004, 2016. 1, 3