SpatialSense: An Adversarially Crowdsourced Benchmark for Spatial Relation Recognition

Kaiyu Yang
Princeton University
kaiyuy@cs.princeton.edu

Olga Russakovsky
Princeton University
olgarus@cs.princeton.edu

Jia Deng
Princeton University
jiadeng@cs.princeton.edu

1. Mapping the Predicates from VRD and VG to SpatialSense

In section 4.1, in order to make the three datasets comparable, we map the spatial predicates in VRD and Visual Genome to their equivalents in SpatialSense. Here we describe the detailed mapping in Table A.

<table>
<thead>
<tr>
<th>SpatialSense</th>
<th>above</th>
<th>behind</th>
<th>in</th>
<th>in front of</th>
<th>next to</th>
<th>on</th>
<th>to the left of</th>
<th>to the right of</th>
<th>under</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRD</td>
<td>above, over</td>
<td>behind, stand behind, sit behind, park behind</td>
<td>in, inside</td>
<td>in the front of</td>
<td>sleep next to, sit next to, stand next to, park next, walk next to, beside, walk beside, adjacent to</td>
<td>on, on the top of, sit on, stand on, drive on, park on, lying on, lean on, sleep on, rest on, skate on</td>
<td>on the left of</td>
<td>on the right of</td>
<td>under, stand under, sit under, below, beneath</td>
</tr>
<tr>
<td>VG</td>
<td>above, above a, above an, above the, are above, are above a, is above, on top of, over</td>
<td>behind, are behind, are behind a, behind an, behind the, is behind, is behind the, on back of</td>
<td>in, are in, are in a, are in an, are in the, flying in, hanging in, in a, in an, in the, inside, inside of, is in, is in a, is in the, laying in, sitting in, walking in</td>
<td>in front of, are in front of, are in front of a, in front of an, in front of the, is in front of</td>
<td>next to, are next to a, are next to the, beside, is next to the, is next to to the, next to a, next to an, next to the, standing next to</td>
<td>on, are on, are on a, are on the, growing on, hanging on, is on, is on a, is on the, laying on, on a, on a a, on an, are on, on front of, on the, painted on, parked on, printed on, sitting on, sitting on top of, standing on, walking on, written on</td>
<td>are left of, in left, in left side of, left side of, on left, on left of, on left side of, to left, to left of, to left of a, to right of</td>
<td>are to right of, on right, on right of, on right of, right of, right of side of, right of side of, to right, to right of, to right of a</td>
<td>under, are under, are under a, are under an, below, beneath, is under the, under a, under an, under the, underneath</td>
</tr>
</tbody>
</table>
Figure A: In adversarial crowdsourcing (section 3 in our submission), the architecture of the robot is an ensemble of a language-only model and a 2D-only model. The language-only model takes two object names along with the predicate (“microwave oven”, “on”, “counter”), and outputs a score for the relation to hold (1.54). The word embeddings of object names are encoded into 512-dimensional vectors by a gated recurrent unit (GRU) \cite{cho2014properties} of 512 hidden units. The same GRU is shared between the subject and the object. The one hot encoding of the predicate is mapped to a 512-dimensional vector by a linear layer. The three feature vectors are fused by element-wise addition, on top of which a 2-layer fully connected network (with 256 hidden units) outputs the score. For the 2D-only model, linear layers map the object coordinates to 512-dimensional vectors, and others remain the same. The final output is the average of these two models.

Figure B: When classifying the predicates in \textit{VRD-Spatial, VG-Spatial} and \textit{SpatialSense-Positive} (table 1 in our submission), we also have a language-only model and a 2D-only model. The architectures are similar to the robot; but there are three differences: (1) The branch for the input predicate is removed, since the task now is to predict the predicate. (2) The output layers now have dimension 9 instead of 1. (3) The object 2D locations are encoded by bounding boxes.
Figure C: These are the language and 2D baselines for spatial relation recognition (section 5 in our submission), which are also used when quantifying the effect of adversarial crowdsourcing (table 2 in our submission). The architectures are the same as the robot, but the object 2D locations are encoded by bounding boxes (They are annotated in a separate process and therefore not available during adversarial crowdsourcing).
(a) The DRNet [Dai et al., 2017] contains a spatial module and an appearance module, which respectively encode the masks of the bounding boxes and image cropped at the union bounding box into 256-dimensional feature vectors. The spatial module contains a hourglass network [Newell et al., 2016], which we find to perform better than a simple stack of convolutional layers. The appearance module is a linear layer on top of a ResNet18 [He et al., 2016] network. The spatial and appearance features as well as the object name features go through an iterative reasoning module that makes extensively use of weight-sharing; all layers with the same name (e.g. fc4) share the same weights. Unlike in the original DRNet paper, we do not perform iterative updates to the object name features, because they are given as ground truth in our task.

(b) For VTransE [Zhang et al., 2017], the bounding boxes are encoded as in the original paper. Image features are also extracted by a ResNet18 network.

Figure D: The specific instance of DRNet and VTransE we use for spatial relation recognition (section 5 in our submission). The input relation is “microwave oven on counter”. The final output is therefore the score for the predicate “on”.

- microwave
- oven
- word embeddings
- GRU
- (0.42, -0.13, -1.09, -1.21)
- concat
- scaling
- W
- 1.24
- 0.64
- -0.63
- -2.26
- fc
- 2.64
- on

- ResNet18
- bilinear interpolation
- fc1
- (0.14, 0.05, 1.11, 1.05)
- concat
- scaling
- W
- 1.24
- 0.64
- -0.63
- -2.26
- fc
- 2.64
- on

- ResNet18
- 256-d spatial feature
- hourglass network
- spatial module
- 256-d appearance feature
- appearance module
- ResNet18
- fc2
- fc3
- fc4
- fc5
- fc6
- fc4
- fc5
- fc6
- fc4
- fc5
- GRU
- word embeddings
- counter
- 2.54
- 0.41
- 3.65
- -0.35
- 3.24
- above
- behind
- in front of
- next to
- on
- to the left of
- to the right of
- under
- 3.45
- on
- ResNet18
- concat
- fc1
- oven
- microwave 
- GRU
- word embeddings
- counter
- above
- behind
- in front of
- next to
- on
- to the left of
- to the right of
- under
- 2.64
- on

- ResNet18
- 256-d spatial feature
- hourglass network
- spatial module
- 256-d appearance feature
- appearance module
- ResNet18
- fc2
- fc3
- fc4
- fc5
- fc6
- fc4
- fc5
- fc6
- fc4
- fc5
- GRU
- word embeddings
- counter
- 2.54
- 0.41
- 3.65
- -0.35
- 3.24
- above
- behind
- in front of
- next to
- on
- to the left of
- to the right of
- under
- 3.45
- on
References


