Modeling Semantics and Pragmatics of Spatial Prepositions via Hierarchical Common-Sense Primitives

Georgiy Platonov, Yifei Yang, Haoyu Wu, Jonathan Waxman, Marcus Hill, Lenhart Schubert

Introduction

Understanding spatial expressions and using them appropriately is necessary for seamless and natural human-machine interaction. However, capturing the semantics and appropriate usage of spatial prepositions is notoriously difficult, because of their vagueness and polysemy. Although modern data-driven approaches are good at capturing statistical regularities in the usage, they usually require substantial sample sizes, often do not generalize well to unseen instances and, most importantly, their structure is essentially opaque to analysis, which makes diagnosing problems and understanding their reasoning process difficult. In this work, we discuss our attempt at modeling spatial senses of prepositions in English using a combination of rule-based and statistical learning approaches. The models operate on a set of artificial 3D ``room world" environments, designed in Blender, taking the scene itself as an input.

Goal: To develop models for spatial prepositions incorporating some of the pragmatic and background-knowledge aspects of the meaning for spatial prepositions

Our approach is based on the following considerations:

- Even though the range of senses of spatial relations together with the heavy dependence on pragmatic considerations make capturing their meaning with simple mathematical criteria difficult, it is still possible to account for many of the above aspects in a principled way. People's judgments about whether a particular relation holds in a given case can be quite variable; therefore it should suffice to provide models that estimate the probability that arbitrary judges would consider the relation to hold.
- Since the usage of locative expressions is pragmatic, the ultimate success criterion in assessing models of prepositional predicates should also be pragmatic; i.e, in physical settings we often use such predicates to identify a referent ("*the blue book in front of the laptop*") or to specify a goal ("*put the laptop on the table*"), so our models should allow a natural language system to interpret such usages as a human would.
- Facilitating explainability. Each relation is built from a combination of simpler relations, whose value can be retrieved and used to provide a justification for a particular judgement.

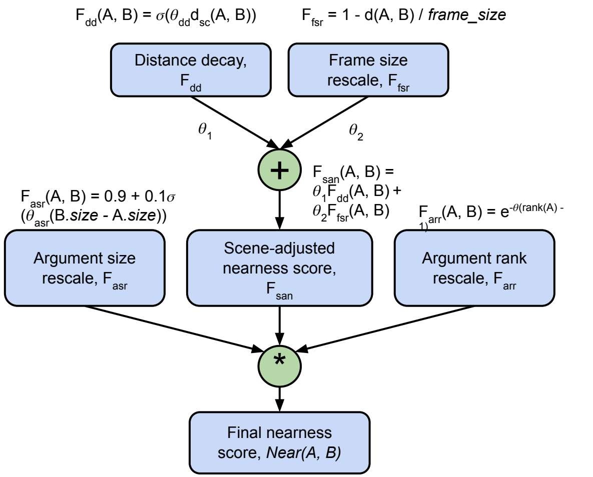
Dataset

We explore spatial prepositions as applied to the so-called ``room worlds" - 3D scenes depicting room interiors filled with common everyday items such as furniture, appliances, food items, etc. The objects in the scene are designed in a particular way, so that their meronomy corresponds to that of the real objects. That is, the mesh consists of parts that are usually distinguished by people (e.g., for a chair, its seat, legs, back, ets., are separate objects that can be accessed by our system). This is useful for part-based inferences, e.g., a book is on a bookshelf when it is on one of the shelves. The objects are also annotated with other additional tags such as frontal vectors that indicate where the ``front" of an object is, object type, etc. We have designed 52 scenes containing about 10-30 objects admissible for annotation as figure objects. Since our annotation task involves describing the location of a figure object in relation to other objects (grounds), objects that form the environment (walls, ceiling, floor) are not admissible as figures (however, they can be used as grounds as in "*the poster is on the north wall*").

Our models

We have developed two kinds of models. The first one is a series of simple multi-layer perceptrons (one per each relation), and the second is our main soft rule-based model, which is implemented as a network (more precisely, an arborescence) of nodes that compute meaningful hand-crafted relations used for determining the values of the prepositions. We rely on an imagistic scene representation for computing spatial relations. Each spatial preposition is implemented as a binary or ternary probabilistic predicate computed hierarchically as a combination of more primitive relations that we call **factors**. These factors encode typical more basic relations that affect whether a particular spatial preposition holds. They are usually either different senses of the same preposition or they co-occur with the preposition in most/all configurations that license the usage of that preposition. The set of factors ranges from those computing geometric properties (e.g., locations, sizes, and distances) to ones computing non-geometric, or functional ones (e.g., physical properties of the relata, such as part structure, or the location of the ``front" of an object). There are several combinatory rules that determine how the factors are combined to produce a composite value. Typically, the factor values are linearly combined, multiplied together, or the maximum among them is taken, depending on the relation. Each node realizes one or more differentiable operations which allows us to train the model using standard gradient descent-based optimization. Each model is essentially a binary classifier used to predict the likelihood that a particular relation holds between given objects.

Example of a factor network



$F_{dd}(A, B) = \sigma(\theta_{dd} d_{sc}(A, B))$

$F_{fsr} = 1 - d(A, B) / \mathit{frame\_size}$

Distance decay, F_{dd}
Frame size rescale, F_{fsr}
θ_1
θ_2
+
Argument size rescale, F_{asr}
Scene-adjusted nearness score, F_{san}
Argument rank rescale, F_{arr}
*
Final nearness score, *Near(A, B)*

$F_{san}(A, B) = \theta_1 F_{dd}(A, B) + \theta_2 F_{fsr}(A, B)$

$F_{asr}(A, B) = 0.9 + 0.1\sigma(\theta_{asr}(B.\mathit{size} - A.\mathit{size}))$

$F_{arr}(A, B) = e^{-\theta(\mathrm{rank}(A) - 1)}$

Evaluation results

Overall, both models performed reasonably well, apart from the cases such as *in front of*, *behind* and *touching* where the rule-based model performed better thanks to additional available information. The results clearly show that it is possible to produce reasonable judgments for most spatial relations even with purely geometric information. However, our main goal was to demonstrate that even when they fall short, our rule-based models still compare reasonably well with pure neural network-based approaches, with the added benefit of being interpretable thanks to their formulation in terms of meaningful decision criteria that correspond to human intuitions about spatial relations.

Relation	Total instances	Accuracy (NN/RB)	Precision (NN/RB)	Recall (NN/RB)	F1 Score (NN/RB)
To the right of	214	0.94 / 0.94	1.00 / 0.97	0.89 / 0.92	0.94 / 0.94
To the left of	152	0.89 / 0.95	0.85 / 1.00	1.00 / 0.90	0.92 / 0.95
In front of	127	0.73 / 0.85	0.66 / 0.81	0.90 / 0.93	0.76 / 0.87
Behind	97	0.76 / 0.86	0.68 / 0.80	0.91 / 0.91	0.78 / 0.85
Above	74	1.00 / 0.90	1.00 / 1.00	1.00 / 0.85	1.00 / 0.92
Below	86	0.82 / 0.87	0.92 / 0.97	0.80 / 0.78	0.85 / 0.87
Between	220	0.96 / 0.95	1.00 / 1.00	0.93 / 0.87	0.96 / 0.93
Next to	331	0.97 / 0.95	0.97 / 0.94	1.00 / 1.00	0.98 / 0.97
Touching	82	0.76 / 0.99	0.74 / 1.00	0.83 / 0.97	0.78 / 0.98
Near	296	0.90 / 0.93	0.91 / 0.95	0.95 / 0.93	0.93 / 0.94
On	346	0.8 / 0.89	0.81 / 0.94	0.89 / 0.88	0.85 / 0.91

Justifying the system's judgments

The main reason for using the rule-based approach is its interpretability. Specifically, our tree-of-factors implementation of spatial models allows backwards-generated justification of the final judgment. Each factor represents some higher-level semantic concept which can be readily translated into natural language. The tree of factors computed during the forward computation phase is preserved and is traversed in the backward direction starting from the root (representing the final output, i.e., the result of the evaluation of the preposition model).

```
For each node in the tree,
depending on the combination rule:
max:
    if current node value >= 0.5:
        return max child
    else:
        return all children
product:
    if current node value >= 0.5:
        return all children
    else:
        return min child
linear combination:
    if current node value >= 0.5:
        return max contributing child
    else:
        return max-weight child
```