Grounded Parsing of Object Attributes and Prepositions
CS 288 Final Project

Yangqing Jia, Trevor Owens and Sergey Karayev
{jiayq,owenst,sergeyk}@berkeley.edu

Spring 2010

Abstract

High-level computer vision and natural language processing are thoroughly intertwined, with the potential to jointly improve performance. We propose a well-defined subset of this under-explored overlap of problems, centered around improving grounded parsing of text and object recognition in images for related pairs of images and text descriptions. We gather a new dataset and present a parsing algorithm to extract object attributes and relations from natural descriptions of images. We evaluate our performance and visualize object co-occurences and prepositions using ground truth data and an annotated set of images. Our results are highly encouraging, and inform directions for further work.

1 Introduction

High level tasks in computer vision such as object recognition are thoroughly related to language. At the same time, tasks in language, such as parsing a paragraph of text, benefit from contextual knowledge that can be obtained from images. We define and study a subset of this under-explored overlap between vision and natural language processing: improving performance of parsing text descriptions of images of indoors scenes. A new dataset for this task is presented and analyzed, object co-occurences and general prepositions are visualized on image data, and performance is reported on a parsing task.

We assume a generative process that leads from the "reality" of a scene to a static photograph of it, and then to a text description of the scene based on that photograph. As illustrated in Figure 1, it is possible to obtain structured information about objects, their attributes, and their relations from two sources: the image itself, or its text description. State-of-the-art computer vision systems are not able to boast of robust performance on this task, with the recognition of relations between objects a particularly unexplored area. Neither are NLP methods robust to the problem, for it is a subset of the general semantic parsing problem, one of the hardest challenges in parsing.

The high-level goal of the line of work we set out on in this paper is in improving the object recognition in images using the parsing of associated text descriptions, and improving the parsing of text with object recognition in images. This dual task is bounded by the "Robot Tour" box in Figure 1, so named for an ultimate demo of this line of work—taking a camera through a cluttered scene and describing relevant objects in natural language, with the object recognition system benefitting from the description. A guiding idea is that object recognition from the image

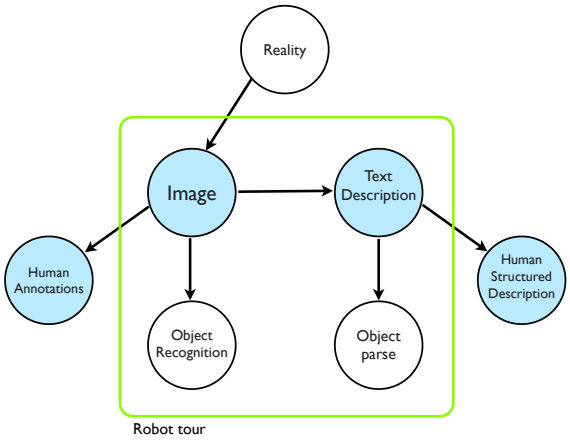


Figure 1: An illustration of the independence assumptions that situate our proposed Robot Tour task.

and object descriptions from the text are dependent on the same set of latent entities. We expect that explicit modeling of this shared latent space will improve performance of both tasks.

In this paper we focus on a small subset of the problems emerging from the high-level goal: using text descriptions of the images to help object detection and classification. Specifically, we propose algorithms to extract object attributes and prepositional relationships by parsing the descriptions of images and to learn the grounded semantics of the prepositions. We discuss the results in terms of accuracy of parsing and intuitive interpretation of our visualizations. The learned semantics of the prepositions can be used to help object localization and further object detection, which we will explore in the future work.

2 Background

Recent computer vision research has started to go beyond simple object classification and to infer plausible attributes of objects, such as color, parts, shape and materials. Farhadi et al [1] treated attribute learning as classification problems, and learned such attributes using labeling information from Mechanical Turk. Lampert et al [2] used low-level text attributes of animals to perform transfer learning, where the attributes for each animal were manually collected in a prior paper. Besides these works which do not consider extracting attributes from text, some work such as [3] attempted to extract attributes such as color and shape from text descriptions of images containing certain objects. Barnard et. al. [4] focused on the opposite direction that considers converting images to text keywords. Several previous works aimed attemptd a multi-modal approach by utilizing image and surrounding documents to help object and scene classification [5] or to help word sense disambiguation [6].

However, all these papers above did not fully utilize the structured information of the text: they either treated the text as a pure source of attribute words, or considered the text as a bag of words, which was converted to a high-dimensional vector that helped classification. Thus, most of the work did not involve extracting attributes of multiple objects in a large paragraph describing

the image, or prepositional relationships between the objects in the scene. In terms of grounded semantic composition for visual scenes, the most relevant paper to ours is [7], where the authors explored the ways people describe scenes, and learned the semantics of several spatially referential expressions. However, their work is carried out on controlled artificial scenes of a cone-world with cones of different colors and positions, and are confined in the interest of learning the text side semantics.

3 Data Collection and Analysis

As reviewed in section 2, there is no dataset that matches our task. There are datasets that pair images of objects with adjectives [1, 8], but not in the context of overall scene. A notable exception is the LabelMe database [9], described in the following subsection, which lists nouns but not adjectives. There is also ongoing research that aims to capture the meaning of prepositions (at MIT Media Lab, for an example [10]). However, we decided to tailor a dataset to our specific requirements. To minimize cost while still gathering data quickly, we decided to leverage an existing dataset for images, while augmenting it with data gathered on Mechanical Turk.

3.1 Image Collection

The LabelMe database[1] contains gigabytes of images from various scenes ranging from indoor office scenes to outdoor streets and natural scenes. Objects in these scenes are then annotated by online volunteers. In general, the spatial relationships between objects in 3D are hard to interpret when we are only presented with the 2D image. For example, in a street scene, the relationship "the tree is in front of the building" may come up with completely different images based on the viewpoint of the camera. However, in some cases such as the indoor office scene, the relationships are more clear, as most images are taken from similar angles. Also, describing such relationships tend to have more practical value: imaging the case of interacting with a robot: we may tell a robot "Please go get my black cellphone. It is on the desk, near the keyboard and in front of that mug", and the robot attempts to understand the sentence, detect the objects and carry out instruction.

We sifted the database and collected 50 office images[2], samples of which are shown in Appendix A, together with the position of 22 most common objects in the images, respectively:

```
bottle telephone mug mousepad mouse keyboard desk screen monitor wall window floor
pen cpu bookshelf poster book chair tablelamp papercup speaker paper
```

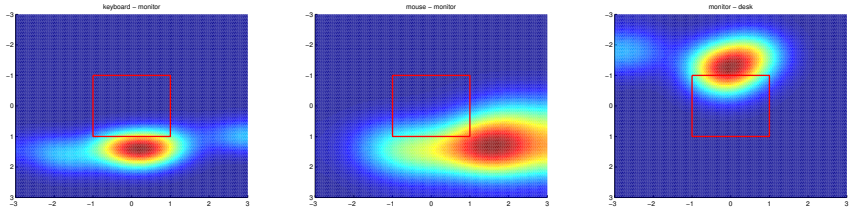
Each object in the image is represented by its bounding box and 2D centroid coordinates, which we will later used to learn the spatial semantics of the prepositions. A total of 460 objects are used in the images, with a mean of 9.2 objects per image and standard deviation 2.5.

3.2 Visualizing Object Co-occurrence

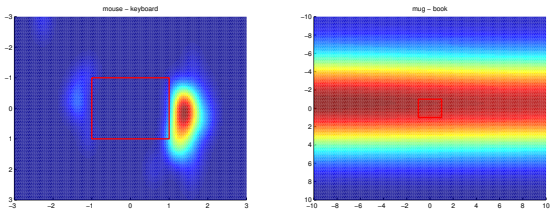
One question is whether object co-occurrence already gives us valuable information about the position of one object given the other, e.g., if we know the position of the monitor, where would

[1] http://labelme.csail.mit.edu/

[2] We should have used a larger set of images, but budget on running Mechanical Turk tasks was too tight for a course project to carry out any large-scale data collections.



(a) keyboard - monitor (b) mouse - monitor (c) monitor - desk



(d) mouse - keyboard (e) mug - book

Figure 2: Conditional probability of the position of the target object given the position of the base object, obtained using kernel density estimation and object co-occurrence statistics. In the subtitles, the first object is the target object and the second is the base object. In all graphs, the red rectangle denotes the normalized bounding box of the base object.

the keyboard be? To answer this we analyzed the co-occurrence of different objects in the dataset. Specifically, we aim to find the conditional probability of the position (defined by the centroid) of a target object O_t, given the position and size (defined by the bounding box) of a base object O_b. To do this, we collected all such co-occurring pairs in each image in the dataset. Then, we normalized the position and scale so that the bounding box of the base object is centered at the origin with edge length 2. The positions of the target object then formed a point cloud around the bounding box. Kernel density estimation (KDE) [11] is applied to infer the probability in the 2-D space, and experimental results on several object pairs are shown in Figure 2.

It is interesting to see that object co-occurrence already gives us a good estimation of the target object in many cases. For example, given the position of the monitor, the keyboard is usually directly under it and the mouse is usually to its lower-right corner in the image. Such properties can also be observed in the mouse-monitor and monitor-desk cases. The mouse-keyboard case actually bears some semantic interpretations: given the keyboard position, the peak of the probability lies to the right of the keyboard, but there is also a smaller peak to the left of the keyboard - possibly due to the fact that left-handed people may place the mouse in that way. The idea that co-occurrence may help us detect objects may also be found in the recent computer vision literature such as [12].

Note that object co-occurrence can be considered as a lower-level utilization of the text information to aid detecting more objects. Take the keyboard-monitor relationship for example, the text "there is a monitor and a keyboard" may well indicate that given the position of the monitor, the relative position of the keyboard should be where most keyboards are. Such information is still primitive, as it heavily lies on the assumption that two objects co-occur often with similar relative positioning in the training images. Given the fact that the number of object pairs, even in the office scene, may grow to a multitude of hundreds or even thousands, the object co-occurrence heuristic

would easily be overwhelmed. The mug-book relationship in Figure 2 is one such example - the only clue we can obtain from co-occurrence is that mugs and books tend to appear at the same height of the image.

However, text description may contain richer information: if we say "there is a mug to the right of the book", ideally we would like to have a probability describing "to the right of", which may look similar to Figure 2d. Thus, learning general prepositional relationships, regardless of what objects they involve, would be a more general and powerful way to exploit text information that helps object detection.

3.3 Natural Language Descriptions of Scenes

Our first text data collection task was collecting descriptions of the images in our dataset. For each of the 50 images in our set, 10 descriptions were obtained on Amazon Mechanical Turk, a web service that allows ordinary people to interact with tasks posted by other people through a convenient web interface, and be paid for their efforts.[3] The "artificial artificial intelligence" of Mechanical Turk is increasingly replacing data collection tasks for which ordinarily a pool of participants would need to be brought into the lab—annotation of images, reaction time tests, or, in our case, describing scenes.

We wanted a fairly complete description of the scene in natural-sounding language, focusing on the objects, their attributes, and relations between them. The task as we assigned it read:

> Given a picture of an indoors scene and a set of objects in it, describe the scene, the objects, and their relations in an informal voice. A list of some objects in the scene with their images is located below the text box. Please refer to them as you write your description. Strive for around 80 words.

The suggested number of words was selected after observing pilot subjects complete our task and was intended to balance the time spent on the task with completeness of description. The mean of the data we gathered was quite close to what we asked for, with mean description of 79 words, with an approximately normal distribution of standard deviation 33.

The quality of the descriptions also varied significantly, but in general was quite satisfactory. As there is no good way to quantify the quality of a description, we present a few sample images, descriptions, and their human parses in Appendix A, with two different descriptions and parses presented for each of three images.

There are three main sources of variation in the different descriptions of the same image. The first is the inherent incompleteness and possibility of what is shown. A biologist could fill pages after a glimpse of a seashell, and a historian could expound at length at the scene of a famous battle, but should their roles be switched, silence would ensue. Given the same boring scene, different people could still describe it quite differently. We attempted to control for it by giving precise instructions on what to include (and implicitly, what not to include), and a target length such that only the most salient information would be submitted. Second, even if two people are constrained to express the same information about the scene, their language will still be different. These first two sources of variation are due to inextinguishable differences between subjects.

The third, most unwelcome, source of variation is due to how different narrators interpreted and followed our directions, and it is an undesirable but sadly inescapable consequence of obtaining data

[3] Amazon Mechanical Turk: https://www.mturk.com/mturk/welcome.

from a cheap source, where we are not able to do training or provide oversight of the participants. As an example, our rules asked the narrators to refer to labeled crops of objects in the scene that we obtained from LabelMe as they described the scene. First of all, our request was ambiguous—were the narrators supposed to use the names of the objects as listed, or just mention all the objects that were displayed, without regard to their (sometimes cryptic) names? Second, even if the request had been unambiguous, our participants would have differed in how well they adhered to the rule. We found this out in the next Mechanical Turk task, for structured listing of objects in the text descriptions.

3.4 Object Parsing of Descriptions

With descriptions of images in hand, we could turn to the next data collection task: collecting the ground truth for evaluating our parser. Our goal for the parser was to derive all objects-attribute relations ("blue mug") and object-preposition-object relations ("the monitor is on the desk") from the paragraph description of a scene. The task as it was posed to Mechanical Turkers read:

> Given a short paragraph that describes a scene, submit all the object relations and object attributes that you find in the text. For relations, pick the one that most closely matches the meaning of the text. Only include the relations that are explicitly given in the text. If there are more than one objects of the same name, number them in the order they appear. Some text is ambiguous–try your best. For attributes, list all the ones that the object is said to have in the paragraph, separated by commas. If an object is not described with attributes, still list it and simply leave the attributes field blank.

The list of prepositions we used was "to the right of, to the left of, in front of, behind, near, above/on top of, below/under, between, to either side of". The prepositions were selected after examining the data—for office scenes, this incomplete list of prepositions seemed to be enough. (One regrettable omission turned out to be 'in'.) It should also be noted that compound prepositions were not possible with this setup, so if a cup was both to the left of and in front of a monitor, the participant had to choose one, or enter both separately.

Number of objects and the number of prepositions was capped at 10. Some of our scenes had more prepositions than that, but we targeted our task to the expected case of just a few objects. The number may have been still too low, introducing undesirable ceiling effects, for the mean number of objects that users parsed from the descriptions was 6.7, with standard deviation 3.3. For prepositions, the mean was 7.0, with standard deviation 2.5. The number of objects with specified attributes was much lower: 2, with standard deviation 2.

For budget reasons, we were only able to collect one parse per description, which was truly regrettable due to the same problems of inherent variation in human interpretation of reality and of the rules of the task that we discussed above. As can be seen in Appendix A, participants once again interpreted the directions they were given with high variance. They were asked to sequentially number multiple objects of the same type, but only a minority actually did. Combined with the ambiguous instruction in the first task to reference LabelMe object names, which were often also sequentially numbered, this led to a significant ambiguity for the evaluation metric, which is discussed in section 5.1.

4 Parsing

The sentences were first parsed using the Berkeley Parser [13, 14]. Each parse tree was sent to an external wrapper algorithm which first discovered any subsentences contained in each sentence using the parse tree. Each subsentence was analyzed separately. The subsentence trees were then used to find the adjectives and associated objects and the prepositions and thier object relations.The method by which the adjectives and prepositional relations were discovered is described below. Another point to make, regarding these parsing algorithms is that they in no way attempt to solve a co-occurence problem. Rather, the algorithms do not distinguish between different occurences of the same word, like *desk*. If you wanted to find attributes in a paragraph of two desks distinctly, there would need to be a preprocessing step which labels these *desk1* and *desk2*, for example. As it stands, we consider each sentence to be an entity of its own, so that this confusion is mostly avoided.

4.1 Object Adjective Discovery Algorithm

As an initial attempt to get some of the simple adjectives, We simply found the adjectives and the nearest nouns to the right in the same noun phrase. This, however, excluded the *adjectival phrases* in one part of a sentence and not the other. For example, in the sentence *The jackelope is large* we have an adjective *large* which is not in a noun phrase and so has no nouns directly to its right. In this case, we can look left for the nearest left branching parent that contains a noun. However, we don't always want the nearest noun to the left, but the subject of the left branching parent. For example, in a sentence like *The jackelope on the rock is large*, we want the noun *jackelope*, not *rock*. What we did to remedy this situation was take the nearest noun to the left that was not in a prepositional phrase, nor a keyword, like *front, side, etc.* This algorithm *buildAdjectiveToObjectMap* is described in Algorithm 1. A supporting algorithm *getSubject* is described in Algorithm 2. This algorithm references another algorithm, *containsPP*, which just returns a boolean if the tree contains a prepositional phrase. In the actual function, for computational reasons, we returned the nested prepositional phrase, which we then searched for prepositions.

4.2 Prepositions

To find the prepositional phrases, we used many of the concepts we discovered in the object attribute discovery. That is, we used the structure of the tree to find the left and right branching parents of the prepositions. We only modeled prepositions that had two objects, as opposed to the more difficult case of prepositions like *between*, which takes three objects. As a simple initial attempt, we found the nouns in the left child of the closest left branching tree and the similarly for the right branching tree. We soon modified this initial idea to take into account many different, important exceptions found in normal language regarding relational specification, such as *The mouse is on top of the desk and to the right of the keyboard.*. In this case, there is a compound preposition *to the right of* which has the first object *mouse* and the second object *keyboard*. Sentences such as these motivate the preposition object finding algorithm. As an example of a way to approach this problem, we will go through the preposition algorithm for the sentence *The mug is also to the right of the keyboard*. The parse for this sentence is pictured in Figure 4. The first step is to find the prepositions, which is easy, since they are in prepositional phrases and are labeled as either TO or IN. For the preposition *to*, we are in the left side of a prepositional phrase, according to the parse

Algorithm 1 buildAdjectiveToObjectMap

```
{Get preterminal trees with label "JJ"}
{Create Hash Map of Hash Maps: nounHasAdjectives}
for preTerminalTree : (preTerminalTrees.label == JJ) do
  adjective ← preTerminalTree.child(0)
  parentTree ← parent(preTerminalTree)
  {Now climb tree to appropriate parent}
  while parentTree ≠ ROOT do
    parentTree ← parent(parentTree)
    if parentTree.getLabel() == NP then
      noun ← getSubject(parentTree)
      nounHasAdjectives(noun,adjective)
      break
    end if
    if parentTree.getLabel().contains(S) then
      leftChild ← parentTree.child(0)
      noun ← getSubject(leftChild)
      nounHasAdjectives(noun,adjective)
      break
    end if
  end while
end for
```

Algorithm 2 getSubject

```
{Create Set: nouns}
nouns ← getNouns(tree)
if containsPP(tree) then
  for subTree : tree do
    if subTree.getLabel() == PP then
      nounsToRemove ← getNouns(subTree)
      nouns.removeAll(nounsToRemove)
    end if
  end for
end if
nouns.removeAll(keyWords)
return nouns
```

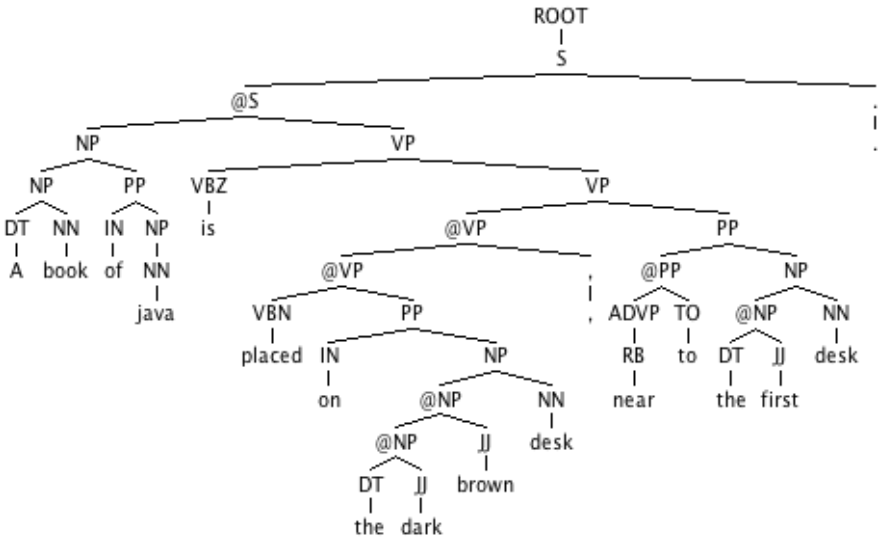


Figure 3: Parse of a typical sentence specifying object positional relationships

tree. However, this is actually a compound preposition, as it is a part of the relational phrase *to the right of*. The preposition of is part of another nested prepositional phrase. We can find this phrase by checking the prepositional phrase that is the ancestor of *to* for a nested prepositional phrase. If one exists, we take all words from *to* to of. We added a step to this to make sure this is a relational phrase, by checking to see if it includes words like *right*, *side*, etc. If it does not contain one of these words, it is rejected. After finding the phrase *to the right of*, to find the object to the left, we find the tree which contains this phrase. In this case it is the prepositional phrase whose word yield is: *also to the right of the keyboard*. If we climb the tree to the first left branching parent and take the left child, we get is. Hence, we ignore this parent and climb further, until we get the left child with yield *The mug*. We can extract all nouns that are not a relational word or a noun, and thus arrive at *mug*. Now to find the second (or right) object of this phrase, we want to start at the last word in the phrase, of. This is inside a prepositional phrase and thus has an object in a noun phrase, *keyboard*. This is typical of most prepositional phrases and motivates the prepositional object finding algorithm. The preposition *to* also turned out to be useful as it often follows a relational phrase, such as *near to, close to, or next to*. In Algorithm 3 we present a high level overview of the algorithm, with implicit reference to many other helper functions. The functions *getLeftObject* and *getRightObject* perform functions similar to those described above.

5 Evaluation

To evaluate our data and our parser, we used the structured object-attribute and object-relations parses from Mechanical Turk as ground truth data for our structured object parses of the descriptions, and to to visualize what prepositions "look like" in the source images.

Algorithm 3 buildObjectPrepositionObjectMap

```
{Get preterminal trees with label "IN" or "TO"}
sentence ← tree.getTerminalYield()
{Create List of Lists objectPrepositionObject}
for i= 0 to preTerminalTrees.size() do
  preTerminalTree : preTerminalTrees.get(i)
  if preTerminalTree.label == IN | TO then
    preposition ← preTerminalTree.child(0)
    parentTree ← parent(preTerminalTree)
    {Now climb tree to appropriate parent}
    parentTree ← getMatchingParent(parentTree,"PP")
    if preposition == TO then
      previousWord ← preTerminalTrees.get(i-1).child(0)
      if previousWord ∈ keyWords then
        preposition ← previousWord + preposition
      end if
    end if
    parentTreeLeft ← parentTree
    parentTreeRight ← parentTree
    if containsPP(parentTree) then
      nestedPP ← getNestedPP(parentTree)
      prepositionNext ← getPreposition(nestedPP)
      j ← getWordIndex(sentence.subList(i+1,end)
      if containsKeyWord(sentence(i+1,j) then
        preposition ← preposition + sentence(i+1,j)
        if preposition == TO then
          parentTreeLeft ← getTopTreeForSpan(i-1,j)
        else
          parentTreeLeft ← getTopTreeForSpan(i,j)
        end if
        parentTreeRight ← getTopTreeForSpan(j,j)
        i=j
      end if
    else
      if preposition == TO then
        getTopTreeForSpan(i-1,i)
      end if
    end if
    parentTreeLeft ← getMatchingParent(parentTreeLeft,"PP")
    parentTreeRight ← getMatchingParent(parentTreeRight,"PP")
    rightObject ← getRightObject(parentTreeRight)
    leftObject ← getLeftObject(parentTreeLeft)
    objectPrepositionObject ← [leftObject,preposition,rightObject]
  end if
end for
```

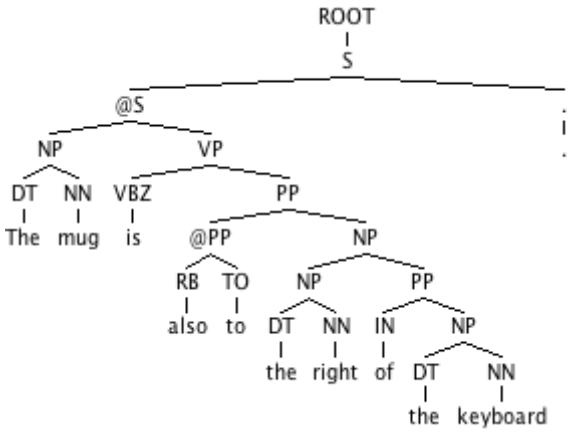


Figure 4: Parse of a typical sentence specifying object positional relationships

Algorithm 4 getMatchingParent

```
{label is the input label to match}
resetParentTree ← parentTree
while parentTree ≠ ROOT do
  parentTree ← parent(parentTree)
  if parentTree.getLabel() == label then
    return parentTree
  end if
end while
return resetParentTree
```

5.1 Dealing with Noise

As mentioned before, each of our 50 images received 10 descriptions (we actually ended up with 504 descriptions total due to some Mechanical Turk peculiarities). Each one of these descriptions was parsed for object-attribute and object-relation information by a human participant. Unfortunately, we obtained only one ground truth parse per description, which made our ground truth data rather unreliable. There is always a trade-off between good data and cheap data, and we evidently made the wrong trade.

As discussed in section 3.3, there are two different types of variability in our data: unavoidable variability due to the inherent ambiguity of human interpretation of the world, and avoidable variability due to differing levels of adherence to the task rules. The latter type would ideally be controlled for during the data-gathering procedure, with training and strict oversight. Since we obtained our data in an uncontrolled setting with participants who were paid just slightly over one dollar per hour of work, the data we received was heavily affected by this second type of variability. To deal with the additional level of noise, we elected to post-process the data.

As a first post-processing step, we ran a manually assembled spelling corrector for some common undesirable spellings of certain object types and prepositions, such as "key board" instead of "keyboard" and "infront" instead of "in front". This allowed the parser to capture adjectives, nouns and prepositions that it otherwise would have missed. Once again, we highlight that had we the means for a high-quality data collection procedure, this additional processing would not have been needed.

Our evaluation metric inherently relied on another type of processing, which attempted to deal with the ambiguity of description of objects and prepositions. For example, a human parser could parse the sentence "there is a mug to the right of the monitor and another cup to the left of the screen" in several ways, from saying that "mug1 is to the right of the monitor" and "mug2 is the left of the screen" to "mug is to the right of the screen" and "cup is to the left of the screen", to saying that "mugs are to either side of the monitor". To counteract at least some of such ambiguities, we relied on a mapping from words to sets of synonyms, where both words 'mug' and 'cup' map to the same set. This way, if the human parser used a word that differed from what was actually in the sentence, our parse of the sentence would still have a chance of matching it. This type of processing was clearly needed even in the absence of misspellings and multiple names for the same object, as in our parsing task we asked the participants to sequentially number objects in case of multiple occurrence, which would introduce nouns that were not in the original sentences.

The same mapping idea had to be used for prepositions, as we restricted the possible prepositions for our human parsers to only nine, when obviously more kinds were used in the actual description. Therefore, a mapping from all possible prepositions to one of nine was required. Our approach was a regular-expression-based matching of words such as "right" to the canonical preposition "to the right of". The mapping was to only one canonical preposition, so if the description had something like "the cup is to the left and above of the desk", it would be mapped to either "to the left" or "above", or to nothing as an illegal preposition (we investigated the effect of the two policies on performance and found it negligible).

5.2 Parsing Results

Precision and recall numbers were obtained independently for object-attributes and object-relations. A match was defined as a non-empty intersect between the expanded sets for the object and the

Scenes	Attributes		Prepositions	
	Precision	Recall	Precision	Recall
Single	20.63	22.29	61.75	43.12
All	21.21	24.88	58.93	40.61

(a) Counting results with 0 matches.

Scenes	Attributes		Prepositions	
	Precision	Recall	Precision	Recall
Single	57.68	62.32	72.33	50.52
All	50.91	59.73	70.88	48.84

(b) Not counting results with 0 matches.

Table 1: Performance of our parser on our dataset.

adjective in the object-attribute element, or for both objects and the preposition in the object-relations element. Matching elements were removed from further consideration to avoid gaming the metric.

Detailed results are given in Table 1. We first ran on the 123 descriptions corresponding to 12 images that contained only one instance of each type of object (i.e. no more than one monitor per image). The performance on this set is less likely to be affected by our mapping, because the objects should not have been numbered at any point in the data collection. We also ran on all 504 parses in our dataset, with broadly similar results. We report mean precision and recall numbers.

5.3 Learning the Semantics of Prepositions

In our second mechanical turk task we obtained a set of prepositional relationships extracted from the textual descriptions by human. We carried out experiments similar to the object co-occurrence analysis to learn the semantics of each preposition we used in the second mechanical turk task. This is done in a similar way to what we used to visualize the object co-occurrence in section 3.

Specifically, for each preposition we used, we collected all the textual triples (target object, prep, base object), and then obtained the object coordinates by pairing them with the objects annotated in the name, using the object name as the pairing criterion. A manual synonym table is constructed to handle synonyms and spelling errors in the data, so that "monitor" and "screen" are considered the same, so are "keyboard" and "keyborad". Ideally, an automated pairing algorithm using synonyms learned from text corpora and robust spelling corrections may benefit if we encounter with a larger set of objects, but a manually written table serves well in our limited case.

One problem in our current learning scheme is that we are not able to distinguish between multiple objects with the same name in the same image: if we encounter "there is a mouse in front of the monitor" but there are two monitors in the image, our current algorithm is not able to distinguish them. To ensure that the learned semantics are correct and to suppress ambiguity, we discarded such triples where one mapping (or both) of the objects between the text and the image cannot be determined. For the nine prepositions we used in the second mechanical turk, a mean number of 124 triples are found for each preposition, with the largest number being 301 (above/on top of), and the smallest number being 7 (between).

We note here that there does exist clues in the text that may help distinguish multiple objects, one important of which would be the adjectives we recovered from the text. However, this requires us to explicitly learn the visual attributes corresponding to the adjectives, which we are planning to implement. Another possible method is the EM approach, where we alternatively learn the semantics of the preposition and the likelihood of choosing one object as the referred one.

After obtaining the object coordinates for the available triples, the conditional probability of the target object given the preposition and the normalized position of the base object is estimated

using the same manner as the one used in visualizing object co-occurrence in section 3. Learned probabilities for all nine prepositions used in our experiment are shown in Figure 5.

It can be observed that in the office scene, the semantics of the prepositions are actually much clear, such as the prepositions "above" and "below". Also, the prepositions "on top of" and "behind" are largely similar to "above" and "below" respectively, due to the common view angle from which we take pictures of the office scene.

However, ambiguity still exists for some of the prepositions, the most notable one being "to the left of". In addition to the high-probability peak that is to the left of the normalized bounding box, there is a lower peak to the *right* of the bounding box too. To explain such ambiguity, we looked into the triples that involves the preposition, and found that a large proportion of such relationships are "something to the left of monitor". we infer that psychologically people tend to personify the monitor when describing directions, which makes the left of the monitor the right side in our view angle, given the fact that monitor faces are usually placed against us.

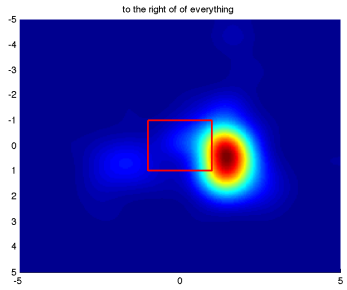
One may raise the question why such phenomena is not as distinctive in the case of "to the right of". This may be explained by the data we collected: a large proportion of relationships involving "right" does not involve monitors but keyboards, books and other objects whom people do not tend to personify. Another possible reason might be that people tend to describe objects in the scene in a certain way - from the left to the right in the image. Thus, when there is a mug to the right of (in the sense of the image) the monitor, we may tend to describe the monitor first and say "the mug is to the left of the monitor" due to personification, instead of describing the mug first and saying "the monitor is to the left of the mug". An evidence supporting this reason is the preposition "near": it is more similar to "right" than to "left", indicating that we may subconsciously tend to describe the objects in the image from left to right.

6 Conclusion

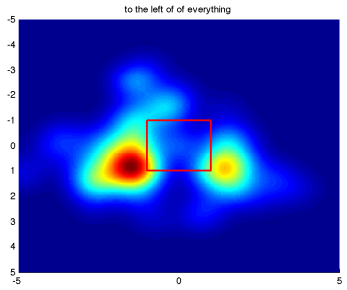
In this project, we focused on two problems involved in our high-level goal of jointly improving object recognition in images and parsing of text descriptions. First, we proposed methods to extract object attributes and their prepositional relationships from text. Second, we learned grounded semantics of prepositions using annotated images. Experimental results showed encouragingly high performance on attributes and especially prepositions, and plausible grounded semantics of the prepositions.

There are several further issues to address. Other than further improving the current learning scheme, the most interesting issue to address is in how to utilize the grounded semantics learned in our experiments to aid object localization and detection in a Robot Tour scenario. Other possible future work includes using the object detection and visual attributes to help parsing the text. Although most sentences are parsed correctly, PP-attachment ambiguity in sentences such as "there is a monitor on the desk behind the keyboard" is still difficult, if not impossible, without the aid of visual information.

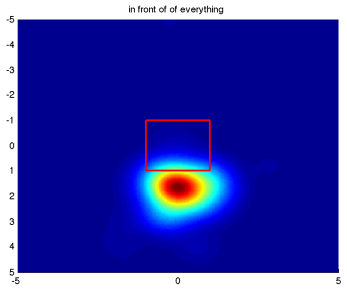
Jointly improving the performance of object detection and sentence parsing is also an interesting problem to approach from the perspective of a shared latent space. For example, we could consider finding a shared latent space which generates both the images and the sentences, as well as independent private spaces for them. We could then attempt to answer the question that is usually assumed to false: do images and text essentially convey the same information, and if false, what information is shared?



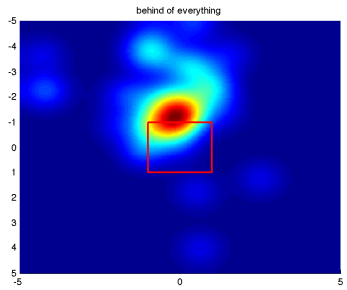
(a) to the right of (253)



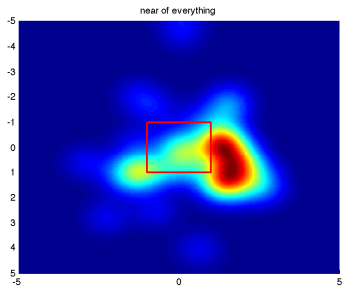
(b) to the left of (132)



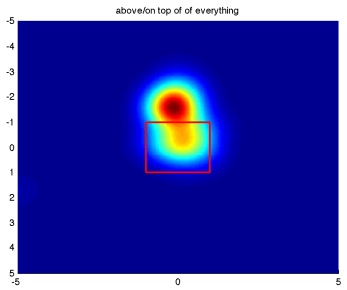
(c) in front of (253)



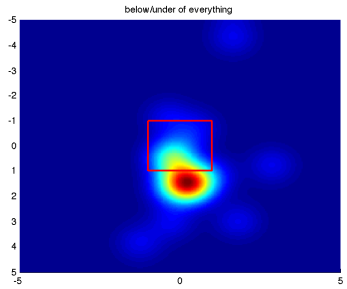
(d) behind (49)



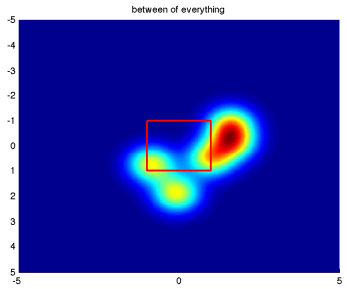
(e) near (57)



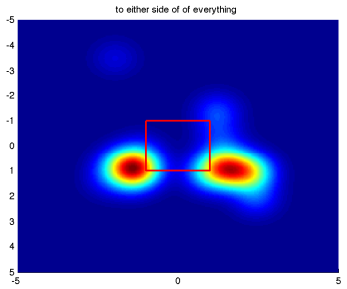
(f) above/on top of (301)



(g) below/under (26)



(h) between (7)



(i) to either side of (42)

Figure 5: The learned distributions for different prepositions, where the bounding box of the base object (the object in PP) is normalized. The number of data points used to generate the distribution is shown in the parentheses.

References

[1] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," *CVPR*, 2009.

[2] C. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," *CVPR*, 2009.

[3] J. Wang, K. Markert, M. Everingham, and U. Leeds, "Learning models for object recognition from natural language descriptions," in *the 20th British Machine Vision Conference (BMVC2009)*, 2009.

[4] K. Barnard, P. Duygulu, and D. Forsyth, "Recognition as translating images into text," *Internet Imaging IX, Electronic Imaging*, vol. 2003, 2003.

[5] A. Quattoni, M. Collins, T. Darrell, and C. MIT, "Learning visual representations using images with captions," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pp. 1–8, 2007.

[6] K. Saenko and T. Darrell, "Unsupervised learning of visual sense models for polysemous words," in *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada, to appear*, 2008.

[7] P. Gorniak and D. Roy, "Grounded semantic composition for visual scenes," *Journal of Artificial Intelligence Research*, vol. 21, no. 1, pp. 429–470, 2004.

[8] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," *Computer Vision and Pattern Recognition.*, 2009.

[9] A. Torralba, B. C. Russell, and J. Yuen, "Labelme: online image annotation and applications," *MIT CSAIL Technical Report*, 2009.

[10] S. Tellex, "Natural language and spatial reasoning," *PhD Thesis Proposal*, 2009.

[11] Z. Botev, "Nonparametric density estimation via diffusion mixing," *The University of Queensland, Postgraduate Series, Nov*, 2007.

[12] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[13] S. Petrov and D. Klein, "Improved inference for unlexicalized parsing," *Proceedings of NAACL HLT 2007*, pp. 404–411, 2007.

[14] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 433–440, 2006.

A Sample data

A.1 First scene

This is an office scene. There is a white computer monitor with a black screen on the desk. There is a white keyboard in front of the computer monitor. To the right of the keyboard is a mouse on top of a blue mousepad. There is a speaker to the left of the computer, and another to the right. There is a book directly to the left of the computer monitor.

(a) First scene (b) Second scene (c) Third scene

Figure 6: Examples of our dataset.

```
('white computer monitor with a black screen', 'white, black')
('white keyboard', 'white')
('blue mousepad', 'blue')
('white computer monitor with a black screen', 'above/on top of', 'desk')
('white keyboard', 'in front of', 'computer monitor')
('mouse', 'to the right of', 'keyboard')
('mouse', 'above/on top of', 'blue mousepad')
('speaker', 'to the left of', 'computer')
('speaker', 'to the right of', 'computer')
('book', 'to the left of', 'computer monitor')
```

> This is a scene of cabin of an office with a computer, that is put on the desk that is round at inside rim,two speakers are put on the desk, speaker1 near a book on which some blank paper 0n it, speaker2 is near the switch board,the computer has CRT sereen with a keyboard near it,a mouse with mouse pad also near the keyboard.

```
('desk', 'round')
('paper', 'blank')
('computer', 'CRT screen')
('computer', 'above/on top of', 'desk')
('speakers', 'above/on top of', 'desk')
('speaker', 'near', 'book')
('blank paper', 'above/on top of', 'book')
('speaker', 'near', 'switch board')
('keyboard', 'near', 'computer')
('mouse', 'near', 'keyboard')
```

A.2 Second scene

This is an office scene.There are two speakers to the right and left side of the monitor.There isa mouse to the right side of the monitor.The desk is situatesd in front of the wall.The telephone is to the right side of the mousepad.The mouspad is situated under the mouse.The keyboard is infront of the monitor.The screen is not working at this time.The papercup 1 is to the left side of the speaker.The papercup2 is to the top of the cabboard.

```
('screen', 'not working')
('two speakers', 'to either side of', 'the monitor')
('mouse', 'to the right of', 'the monitor')
('desk', 'in front of', 'wall')
('telephone', 'to the right of', 'mousepad')
('mousepad', 'below/under', 'the mouse')
('keyboard', 'in front of', 'monitor')
('papercup 1', 'to the left of', 'speaker')
('papercup2', 'above/on top of', 'cabboard')
```

A ROOM IS SHOWN WHICH HAS A COMPUTER MONITOR ON THE DESK. THE SCREEN OF THE MONITOR IS BLANK. A KEYBOARD LIES IN FRONT OF THE COMPUTER. THERE ARE TWO MICE ON THE DESK. ONE MOUSE IS KEPT ON A MOUSEPAD AND THE OTHER MOUSE IS IN FRONT OF THE SPEAKER. THERE IS ALSO A BLACK TELEPHONE ON THE DESK. A PAPERCUP IS ON THE DESK AND THE OTHER PAPERCUP IS ON TOP OF THE ALMIRA.

```
('screen', 'blank')
('black telephone', 'black')
('monitor', 'above/on top of', 'desk')
('keyboard', 'in front of', 'computer')
('mouse1', 'above/on top of', 'mousepad')
('mouse2', 'in front of', 'speaker')
('black telephone', 'above/on top of', 'desk')
('paper cup1', 'above/on top of', 'desk')
('paper cup2', 'above/on top of', 'almira')
```

A.3 Third scene

This is an office scene with two desks. The first desk has a computer monitor on it with a keyboard in front of it and a mouse with a mousepad to the right. Directly behind the mouspad there is a speaker and below the mousepad on a bottom shelf there is another keyboard. There is a grey swivel office chair in front of the desk. Beyond the first desk there is a series of bookshelves. The second desk has three water cups on it along with some books and paperwork.

```
('chair', 'grey, swivel, office')
('water cups', '3')
('monitor', 'above/on top of', 'desk1')
('keyboard', 'in front of', 'monitor')
('mouse', 'to the right of', 'monitor')
('speaker', 'behind', 'mouse pad')
('book', 'above/on top of', 'desk2')
('shelf', 'below/under', 'mouse pad')
('keyboard2', 'above/on top of', 'shelf')
('chair', 'in front of', 'desk1')
('bookshelves', 'near', 'desk1')
('water cups', 'above/on top of', 'desk2')
```

A multiple functional desk with more than one person viable. There are three viable computer stations with a book row behind the desks. The desks are unorganized with many cans and mugs sitting around also books, papers, folders, earphones, and many other items sit on the desks. A dry erase board with writing on it sits behind the computers.

```
('desk', 'multi functional')
('desks', 'unorganized')
('erase board', 'dry')
('row of books', 'behind', 'desks')
('cans', 'above/on top of', 'desks')
('mugs', 'above/on top of', 'desks')
('books', 'above/on top of', 'desks')
('papers', 'above/on top of', 'desks')
('folders', 'above/on top of', 'desks')
('earphones', 'above/on top of', 'desks')
('erase board', 'behind', 'computer')
```