SiRoK: Situated Robot Knowledge — Understanding the Balance Between Situated Knowledge and Variability

Angel Daruna,[1] * **Vivian Chu,**[1] **Weiyu Liu,**[1] **Meera Hahn,**[1]
Priyanka Khante[2] **Sonia Chernova,**[1] **Andrea Thomaz**[2]
[1]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA.
[2]Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA.

Abstract

General-purpose robots operating in a variety of environments, such as homes or hospitals, require a way to integrate abstract knowledge that is generalizable across domains with local, domain-specific observations. In this work, we examine different types and sources of data, with the goal of understanding how locally observed data and abstract knowledge might be fused. We introduce the Situated Robot Knowledge (SiRoK) framework that integrates probabilistic abstract knowledge and semantic memory of the local environment. In a series of robot and simulation experiments we examine the tradeoffs in the reliability and generalization of both data sources. Our robot experiments show that the variability of object properties and locations in our knowledge base is indicative of the time it takes to generalize a concept and its validity in the real world. The results of our simulations back that of our robot experiments, and give us insights into which source of knowledge to use for 31 types of object classes that exist in the real world.

Introduction

Robotics is undergoing a transition from the development of specialized, single-task robots to general-purpose platforms expected to operate in diverse and changing environments, such as hospitals and homes. Operation in unconstrained human environments introduces many new challenges, one of which is that of knowledge acquisition. On the one hand, the diversity of target environments makes it impossible to precode the robot with all the required knowledge (e.g., where the towels are kept, that a particular bowl is made of metal), requiring the robot to learn from observations on-site. On the other, information often referred to as “common sense knowledge”, can be transferred across domains (e.g., towels are often found in bathrooms and closets, bowls are containers) (Speer and Havasi 2012). In this work, we examine different types and sources of such data, to understand how

*This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650044. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

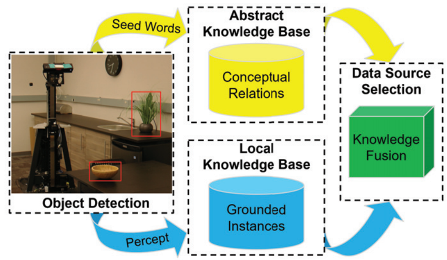

Figure 1: High-level view of SiRok framework.

locally observed data and abstract knowledge can be fused to enable a robot to most effectively reason about its world.

As a motivating example, consider a robot placed in a new home and tasked with fetching a glass of water. One approach is for the robot to rely entirely on local observations, and to exhaustively search the environment for a glass and sink. A human visitor to the home, however, would instead be likely to first find a kitchen, then begin to open cabinets (and not drawers) in order to find the glass. This behavior would be guided by semantic, domain-independent knowledge gathered from prior experiences, and a similar capability would enable robots to more effectively adapt to new environments. However, local knowledge must also be incorporated into this reasoning, allowing adaptation to domain-specific patterns or the current state of the world, such as when the glasses have already been set out on the table, or in houses with unconventional item storage areas. In order to support a robust deployment model, we must better understand the limits of both local and abstract data.

In this work, we consider two sources of knowledge: abstract knowledge and local knowledge. We characterize *abstract knowledge* as domain-independent information that generalizes across many environments (e.g., food in typical homes can be found in the refrigerator in the kitchen). Specifically, we use commonsense information from ConceptNet (Speer and Havasi 2012) and WordNet (Miller 1995) to allow the robot to reason about novel objects and environments. We characterize *local knowledge* as information the robot has perceived in its current environment. This includes information obtained from its sensors (e.g., camera, laser, etc.), including object recognition, semantic locations, and object properties. From these data sources we generate two separate knowledge bases, the Abstract Knowledge Base (AKB) and the Local Knowledge Base (LKB), which the robot uses to reason about the world. Combined, these components make up the Situated Robot Knowledge (SiRoK) framework (Fig. 1).

Our work makes the following contributions. First, we introduce a domain-independent framework for automatically retrieving common-sense knowledge for a given environment. We use object labels, obtained from object recognition, to generate seed words, which are then used to query existing semantic knowledge bases to construct a probabilistic model representing object type, location, and property data. Second, in a series of robot and simulation experiments we examine in what situations the abstract and local knowledge sources are most reliable for objects with both mutable and immutable properties. Our results show that variability is a key heuristic to take into account when evaluating knowledge sources. In particular, as variability increases, we should emphasize sources of general knowledge. For cases with extreme levels of variability, a robot should rely on direct observations or chance. Our simulations validate the trends we see in our robot experiments, and extend our conclusions to 31 different classes of objects found in real-world households.

Related Work

Numerous projects across the AI community have sought to make use of commonsense and semantic knowledge. Three large-scale commonsense knowledge networks used across a wide range of applications are WordNet (Miller 1995), ConceptNet (Speer and Havasi 2012), and ResearchCyc (Lenat 1995; Matuszek et al. 2006). WordNet consists of a collection of synsets, which connect concepts hierarchically through the *IsA* relation. WordNet also distinguishes between different senses of the same word and provides glosses, or definitions, for each sense. While WordNet is clean and hand-coded, it also lacks diversity in the types of relations it contains. ConceptNet, on the other hand, contains several dozen different relations, but it does not distinguish between word senses and is largely crowdsourced, leading to a large amount of noise. ResearchCyc uses an even larger number of relations (currently around 17,000) to connect concepts. For the purposes of this work, we choose to use data from WordNet and ConceptNet to take advantage of the complimentary benefits of each.

In other work, Zhu, et al. (Zhu, Fathi, and Fei-Fei 2014a) perform affordance prediction on a set of images by using a Markov Logic Network (MLN) (Richardson and Domingos 2006a) to represent affordance knowledge. This work also does not deal with context and used hand-selected objects and affordances in the network. In (Chen and Liu 2011), contextual noise is addressed by disambiguating the concepts in ConceptNet to enrich the WordNet senses with more diverse knowledge for improved performance on word sense disambiguation tasks. While disambiguating ConceptNet helped provide context for each of its concepts, the resulting knowledge base contained only abstract information. In contrast to this approach, (Stoica and Hearst 2004) did

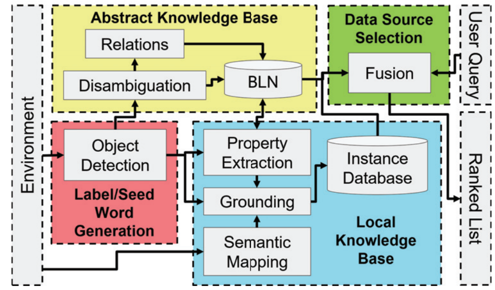

Figure 2: System architecture for the Situated Robot Knowledge (SiRoK) framework. The pipeline starts with environment data that is used to populate the AKB and LKB

construct a situated knowledge hierarchy in a (nearly) automated way, however, the resulting model only included hypernyms (the *IsA* relation).

Within robotics, the KnowRob (Tenorth and Beetz 2009) and RoboBrain (Saxena et al. 2014) projects are most closely related to our work. In KnowRob, the authors create a knowledge network from a variety of encyclopedic sources and represented the network using Prolog rules and the Web Ontology Language. This network is then used to repair robot task plans by filling in missing low-level details from high-level task descriptions. In RoboBrain, the authors generate a multimodal knowledge network for robotics using data collected automatically from the web. The resulting network is abstract and does not account for the domain-specific details relevant to the situational context of the robot. The RoboEarth project focused on the creation of a cloud repository of generalizable robot knowledge, including object models and robot task descriptions, that could be transferred across robot platforms and domains (Waibel et al. 2011). While these works deal with both abstract and situated knowledge, none of them investigate which knowledge source to leverage when. Our efforts focus on understanding which knowledge source a robot should use given some query (e.g. where is the plant) which may be part of a higher-level task. We conclude that the variability of a given piece of information impacts the reliability of obtaining it from either local or abstract sources.

SiRoK System Architecture

The SiRoK framework is implemented as a system of interconnected modules, which communicate using ROS. The system has three main components (Fig. 2): AKB, LKB , and Data Source Selection, each of which contains a series of subsystems that aggregate and process data. At a high-level, the pipeline begins by performing object detection, where objects in the environment are assigned an object class labels (e.g., cups, bowls, etc.). These generated class names become seed words that are used to extract information from online commonsense networks to build an AKB. These object class labels are also used during grounding, where specific object information is stored into the LKB. In Data Source Selection, the robot uses specific queries to ask

Data Types	Possible labels
Object Class	apple, banana, book, bottle, bowl, broccoli, cake, carrot, chair, clock, couch, cup, donut, fork, glass, knife, laptop, microwave, orange, oven, phone, pizza, plant, refrigerator, sandwich, sink, spoon, table, toaster, tv, vase
Colors	black, blue, brown, gray, green, orange, pink, purple, red, transparent, white, yellow
Materials	cloth, glass, metal, organic, paper, plastic, wood
Weights	light, medium, heavy
Shapes	arch, cylindrical, rectangle, spherical

Figure 3: Classes and object data in the AKB and LKB

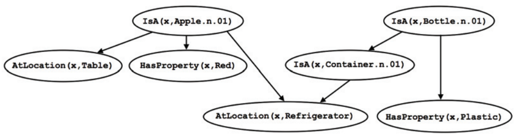

Figure 4: An example of abstract knowledge represented using a Bayesian Logic Network (BLN)

for information from AKB and LKB and fuses the results to respond to the queries. In the remainder of this section, we describe each subsystem in detail and the full system diagram can be found in Fig. 2. The colors of each component in Fig. 2 match the high-level view in Fig. 1.

Object Detection

For object detection, we used the open source real-time object detection system YOLOv2 (Redmon et al. 2016). YOLOv2 uses a convolutional neural network and computes the location and classification of each object in an image in a single pass. It does this by dividing the image into cells, calculating an objectness score and then object classification probabilities over the individual cells, it then using anchor boxes to predict the object bounding boxes. We tested YOLOv2 on PASCAL VOC2012, achieving a mAP (mean average precision) score of 73.4. For our robot experiments, we trained YOLOv2 on the subset of COCO (Lin et al. 2014) object classes which are specific to the home environment (Fig. 3). Each time the system recognizes the object, the object label, bounding box of the object, and raw rectangle segment of the object is sent to the LKB. The object labels are also passed to the AKB.

Abstract Knowledge Base

We represent the robot's AKB as a Bayesian Logic Network (BLN) (Jain, Waldherr, and Beetz 2009), a directed statistical relational model in which the variables under consideration are represented as first-order terms or predicates with arguments. BLNs allow logical constraints, represented as first-order logic rules, to be imposed on the network. Prior work in computer vision has utilized Markov Logic Networks (Richardson and Domingos 2006b), a representation that unifies Markov Random Fields and first-order logic, for modeling object attributes and affordances (Zhu, Fathi, and Fei-Fei 2014b). However, parameter learning in MLNs is an ill-posed problem (Jain, Kirchlechner, and Beetz 2007) and approximate inference is expensive even for simple queries. In contrast, BLNs are easy to train, more efficient and have scaled better to our application. Fig. 4 shows a small example BLN, which, once constructed, can be used to perform inference using likelihood weighting (Fung and Chang 2013) to answer queries such as *AtLocation*($Object_i, x$) or *HasProperty*($Object_i, x$).

To construct the BLN, we leverage information from two online sources of semantic knowledge, WordNet (Miller 1995) and ConceptNet (Speer and Havasi 2012). WordNet is a low-noise hand-crafted collection of sets of cognitive synonyms (synsets), each expressing a distinct concept (e.g., *spoon*) and related to other concepts through hypernym (the *IsA* relation, e.g., *IsA(spoon, utensil)*). ConceptNet is an auto-generated commonsense knowledge bank; it does not differentiate between word senses but groups all within a single concept node related to others through multiple possible relations. For example, for the object *mouse*, ConceptNet returns *AtLocation(mouse,office)* and *HasProperty(mouse, organic)*, highlighting the need to perform sense disambiguation to correctly parse this data.

Given seed words obtained from object recognition labels, we first perform sense disambiguation using the technique in (Tsatsaronis, Varlamis, and Vazirgiannis 2008), by finding the sense of each word that maximizes the overall similarity between the seed words (leveraging the fact that the words come from the same context). We then query WordNet and ConceptNet for semantic data related to each disambiguated word. Importantly, the seeds words not only provide a starting point for data retrieval, but together act as context for the robot's specific environment. Currently, we retrieve data for three relations, which we selected due to their usefulness in robot task execution.

- *IsA*: determines the relationship between an object and its hypernym (e.g., *IsA(bowl, container)*), allowing the robot to reason over object categories.
- *AtLocation*: determines the relationship between an object and locations in the world. (e.g., *AtLocation(bowl, sink)*, allowing the robot to query likely object locations.
- *HasProperty*: determines the relationship between an object and properties such as materials, shape, and colors (e.g., *HasProperty(bowl, ceramic), HasProperty(bowl, red)*, aiding in recognition and allowing the robot to reason about possible object uses (e.g. metal objects should not be placed in the microwave).

For each relation, we calculate a likelihood based on a weighted combination of the relation score from ConceptNet and the Explicit Semantic Analysis relatedness measure (Gabrilovich and Markovitch 2007) between the two concepts in the relation. This likelihood provides an initial estimate for the real-world probability of a given relationship and enables us to generate training evidence for BLN based on the distribution. Relations that cannot be sampled directly are inferred logically using transitive prolog rules. For additional details, see (Garrison and Chernova 2016).

Local Knowledge Base

LKB Data Structure We represent the robot's local environment through a collection of object instances, forming a

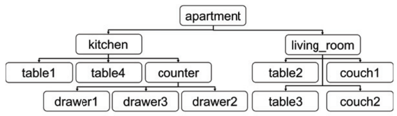

Figure 5: Topological map.

Figure 6: Metric map with an overlay of the spatial volumes associated with nodes in the topological map.

memory of encountered items, and their locations and properties. For $o \in O$, each object class out of the set of objects known to the robot (listed in Fig. 3), we store i instances of that object within the LKB, where an instance is defined as a unique object.

The LKB is implemented using PyTables and HDF5; each object class o is stored as a database, with a table generated for each object instance. For each instance, we currently store the object label, previously seen locations (pose and semantic label), image region corresponding to the bounding box from object recognition, visual information (RGB-D values), and all properties known about the instance (e.g, color, material). The resulting representation provides a scalable memory system that allows for efficient retrieval of all of its recent memories of instances.

Grounding In addition to using object recognition for object class labels (e.g., *bottle*), the robot must distinguish different instances of the same class (e.g., *red bottle* vs *yellow bottle*). The grounding component of SiRoK uses features distinct to instances of an object class to distinguish among multiple instances. This form of grounding, from here on referred to as *instance grounding*, was implemented using a K-Nearest Neighbors (KNN) classifier with a threshold distance to accommodate new instances of a class. Our implementation relies on color properties, extracted from the bounding box region of the image using the GrabCut algorithm (Rother, Kolmogorov, and Blake 2004) and uses KNN to determine whether an object is a new instance. Grounding enables the robot to perform color-based differentiation of objects, which we leverage in our study. In future work, we will expand instance grounding to incorporate spatial and temporal information about objects, as well as a wider variety of features.

Semantic Location In order to effectively generalize local information and relate it to abstract knowledge, we require a method for converting the robot's world coordinates to semantic location labels (e.g., *kitchen counter*). To provide a semantic location for an object, we utilize a hybrid map (Buschka and Saffiotti 2004), which links a topological map, consisting of a tree graph representing human domain knowledge, with a metric map of spatial locations in the environment. Fig. 5 and Fig. 6 show the topological and metric maps used in this work. The links between the topological map and metric map are expressed directly in the topological map nodes; association of each node with a volume in the metric map. This map structure enables the robot to obtain a semantic label for any 3D point that is hierarchical (e.g., object o is in a *drawer* in the *kitchen* in the *apartment*).

Property Extraction As discussed above, SiRoK enables the robot to reason about a range of object properties, including color, weight, material and shape. Through local observation, the robot is able to obtain some properties (e.g., color), while other important object characteristics (e.g., material) are very difficult to determine for existing platforms. Some complementary information, however, can often be obtained from the AKB, which obtains property information through ConceptNet. For each object, we assign a set of object properties commonly learned and used by robots (Hermans, Rehg, and Bobick 2011; Sun, Bo, and Fox 2013; Sinapov et al. 2014). These include color, shape, material, and weight. The individual values that each object can take on (e.g. blue, heavy, metal, etc.) can be found in Fig. 3.

While color is obtained using a simple color classifier, we hand-label the shape and weight of the objects. With the current state of the art we assume that these properties can be obtained easily with good accuracy via existing machine learning algorithms and the use of pre-trained classifiers (Chu, Fitzgerald, and Thomaz 2016; Sun, Bo, and Fox 2013; Sinapov et al. 2014). Future work will include exploration of the objects using the robot's arm and visual information from the RGB-D camera to learn the object properties. However, material still remains to be one of the harder properties to be learned. In this work, we can leverage a human in the environment to extract the material properties of the objects.

In its existing form, the BLN contains far too many property edges to simply verify each one with the human. Thus we present an algorithm, which takes the existing BLN generated from ConceptNet and WordNet, and actively selects a subset of property relations to verify with the human. This results in a pruned representation that is consistent with the specific objects in the current environment.

We first modify the BLN to include inter-property edges. For all properties in the BLN, we add an edge if a relation exists between them in ConceptNet. We then generate three tables. $T_{material}$: all material properties present in our BLN (i.e., holds a relation with *Material* in the ConceptNet). For the next two tables, we use the association index in ConceptNet, a measure between 0 to 1 of how related two words are. $T^{O_N}_{assoc}$: holds all the association indices between an O_N and every property belonging to that object (we ignore properties with index < 0.07).$T_{interprop}$: Let P_O be a set such that each $p \in P_O$ is a property of O, this table holds the inter-property association indices between any two properties in P_O.

Next, we systematically pick the properties to query an expert for verifications. For each object, we query the expert about property, $p \in P_O$ with the highest association index in $T^{O_N}_{assoc}$. If it is verified *true* and exists in $T_{material}$, then all other material properties belonging to that object are assumed to be *false* and are not queried. We can also assume the predecessors of that property are true for O_N (e.g., if Aluminum is true, then Metal can be assumed true). For the successors, we assume their *hasProperty* relations are true (e.g., Metal true, then Opaque true), but need to query the successors with an *IsA* (e.g., if Metal true, still need to ask about Aluminum). If a node in this *isA* set is verified to be *true*, the rest are assumed to be *false*.

Next, query with the a property with the minimum inter-property association index with p, to ask the most different question next. Repeat this process until all the properties are verified as *true/false*. We construct an expert-verified BLN, *vBLN*, with all verified *true* properties. For evaluation we will look to compare this verified BLN with a ground truth BLN with a dissimilarity index, $I_{dissimilarity}$, defined as:

$$\frac{\text{Uncommon edges between ground truth and vBLN}}{\text{Total number of unique edges in ground truth and vBLN}}$$

Data Source Selection

SiRoK uses knowledge from the AKB and LKB to handle object *queries* related either to (1) what the object is, (2) where it is located, or (3) what properties it has. Within the AKB, the BLN is queried for *IsA*, *AtLocation*, and *HasProperty* information, and the results sorted by probability value. The LKB answers *AtLocation*, and *HasProperty* queries by using the stored outputs from semantic mapping and property classification, returning a ranked list of the most frequently encountered property. We note that, in general, location and property information have different characteristics. A specific object is likely to change location, possibly even frequently, whereas most of the properties we consider, such as color, are likely to change less often. Locations and properties also often generalize across instances (e.g., cups of the same color or cups stored in the same cabinet), but this depends on the variability of the object. In the next section, we evaluate how our inference performs across these different data types.

Robot Experiments

To evaluate the SiRoK system and examine the relative applicability of abstract knowledge and local knowledge, we designed a series of experiments testing the robot's ability to predict object locations and properties. Our test environment resembles a simple apartment containing furniture and different use areas, as seen in Fig. 6. For all experiments, we use the robot platform, Prentice (Fig. 1). Prentice is an omnidirectional mobile robot and has a horizontally mounted lidar for navigation and a Microsoft Kinect2 RGB-D camera mounted on a pan/tilt unit for visual sensing.[1]

[1]Note that we do not evaluate *IsA* queries on the robot due to the highly abstract nature of the data. *IsA* results are reported in the simulation section.

Building the Knowledge Bases

We populate an AKB by using the 31 possible class labels shown in Fig, 3 to seed a BLN using ConceptNet and WordNet. As described in *SiRoK System Architecture*, these class labels come from the COCO image dataset that are associated with kitchen and living rooms. We removed one label, hot dog, due to WordNet disambiguating hot dog to sandwich. This is due to WordNet characterizing that hot dogs are sandwiches, which is partially true (i.e., a hot dog is a piece of meat between bread). Future work will address how to take into account words that are part of the same hypernym hierarchy. The constructed BLN contains 257 nodes and 358 edges.

To gather data for the LKB, we used the following experimental steps: (1) put object(s) in our testing environment, (2) allow the robot to observe the environment and update the LKB, (3) update the state of the object(s) in our environment, then repeat this process for the desired number of observations. After each observation, we evaluate the accuracy for finding objects or naming object properties on a fixed test set. To populate the semantic locations, we provide an expert labeled semantic map that correlates to the described scene in Fig. 6. We use a color classifier to label each object in the test environment and the BLN for the object material. The average classifier accuracy is 70% and average clarifications needed for object property is 2.

If a human is available, SiRoK has the option to interactively validate properties in the BLN. We performed 84 clarifications to prune 50 edges in the vBLN from 195 property edges using the human-verification algorithm mentioned in Section III-C.4. While this is a large number of clarifications, during a deployment such queries could occur over a length of time (multiple days) as the robot spends time learning about its environment. Moreover, our algorithm is currently limited by ConceptNet. ConceptNet lacks rich inter-property knowledge (i.e. if an *apple* is sweet, one can assume it is also *juicy*) and the notion of classes (i.e. *sweet, sour, spicy, tangy* all belong to the same class of *taste*), the number of queries is large. However, knowledge of *material* class and good inter-property knowledge, it fared well for *bottle* where only 3 queries were asked for 9 properties or only 1 for 5 properties of *cup*. The final dissimilarity score of the vBLN to ground truth object properties is 0.11 (6 edges difference). This means that the BLN is only 6 edges (an edge is between an object and a property) away from the ground truth and managed to learn 50 out of the total 53 edges from the ground truth.

Experiments

We break down this section into two experiments: (1) finding objects in the scene and (2) determine the properties of objects. For both, we hypothesize that the role of variability in object options is a primary factor in deciding when to use abstract vs. locally learned knowledge. If an object moves around more frequently, we should rely on reasoning about where we might find the object as opposed to remembering where was the last time or most frequently seen location. For object properties, we expect to see a similar trend.

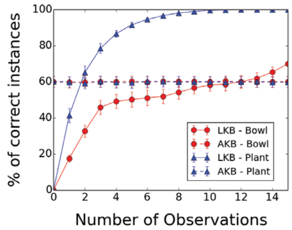

Figure 7: Average accuracy (AKB vs. LKB) across 5000 permutations to predict the top 3 locations of *potted plant* and *bowl*

Finding Objects For object location, we collect two separate sets of observations, one using a *bowl* and one using a *potted plant*. The two objects can be seen in Fig 1. For each object we collected 20 observations of the objects in various locations in the kitchen and living room. We determined the locations for each object using two different distributions for each object, one with more movement and one with less movement. The *bowl* was on the higher end of a variability spectrum (table1: 20%, table4: 20%, counter: 12%, table2: 12%, table3: 12%, drawer1: 12%, drawer2: 12%), while the *potted_plant* object was on the lower end (table3: 50%, table2: 25%, counter: 25%). Each time an object is detected by the robot, the object's semantic location is written to LKB.

To test and compare AKB and LKB, we randomly select 25% of the observations to leave out as the test set. This results in five observations in the test set and 15 in the train set. We test the accuracy AKB and LKB incrementally by introducing each observation separately. Specifically, we ask AKB and LKB to predict the location of the 5 observations in the test set after seeing one observations, two observations, and so on. AKB and LKB predict the locations by providing a ranked list of possible locations as described in *Data Source Selection*. We randomly select 5000 different permutations of the observation order and report the average accuracy and standard deviation to account for orderings effects. Note that the AKB is generated prior to seeing the observations as it represents general domain-free knowledge, so the accuracy of the AKB does not change over observations.

The results of this test can be seen in Fig. 7 where the robot turns the top three locations from its ranked list (simulating if the robot were allowed to look at three different locations to find the object). We can see that for the *potted plant*, the LKB reaches 80% accuracy by the fourth observation. However, for the *bowl*, the overall accuracy of the LKB reaches only 65% for top three locations, which is only slightly better than chance. When comparing AKB to LKB, it is clear that in cases where there is low variability in the current environment, learning about the object's location is superior to using general knowledge. However, for the bowl, where locations are more varied, the AKB does a better job of reasoning where in general might bowls be located. Furthermore, for both cases, when there is little to no knowledge of the scene, AKB still offers some insight to where the object might be located as opposed to LKB. We observed the

Figure 8: The bottle outlined in long green dashes, solid blue lines, and dotted red lines are plastic, metal, and glass respectively. The bottles are colored from left-to-right as blue, pink, green, blue, white, white, yellow, red, green, and green.

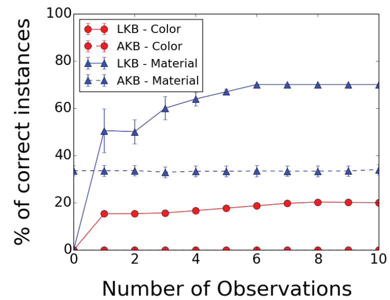

Figure 9: Average accuracy (AKB vs. LKB) across 5000 permutations for predicting the top property of 10 different bottles for two different properties (color and material)

same trends when testing the top-one results, although with lower overall performance rates.

Object Properties As described in *Data Source Selection*, object properties are fixed to a specific instance. As a result, we test the robot's ability to predict object properties by using a fixed test set that is also the observation set. As the robot observes its environment over time (similar to how one gets acquainted to a new environment), all of the objects in the environment will be added to its observation set. We select objects of the same class type (e.g., all bottles), to determine if knowledge properties of specific objects can provide insight on the general class of objects. Similar to object locations, we hypothesize that the variability of possible values for a property affects when and how we use our knowledge base. As a result, we select bottles with varying levels of variance within its properties (i.e., color is highly variable while materials is not). For this specific experiment, we selected 10 *bottles* (Fig. 8). Specifically, they ranged in color (green: 3, blue: 2, white: 2, yellow: 1, red: 1, pink:1) and materials (plastic: 7, metal: 2, glass: 1) with color more variable and material less.

The results of the test across the 10 bottles can be found in Fig. 9 for both color and material. We limit the AKB and LKB to just one guess as opposed to three for locations because for object properties, there is a higher threshold for errors. While searching three different locations in a home environment might take slightly longer, it is not unreasonable or dangerous for the robot to do so. On the other hand, predicting that an object is not metallic and putting it in the microwave could have dire consequences. As expected, the LKB performs poorly at predicting highly varied object properties. This makes intuitive sense as knowing that one

Class	Location				Color				Material				Type
	1		15		1		15		1		15		
	AKB	LKB	AKB	LKB	AKB	LKB	AKB	LKB	AKB	LKB	AKB	LKB	AKB
Apple	13	15	13	63	73	29	73	40	0	100	0	0	Food, Produce, Edible fruit, Fruit, ...
Banana	10	13	10	37	47	36	47	47	0	100	0	0	Food, Produce, Edible fruit, Fruit, ...
Book	0	8	0	23	0	14	0	20	100	100	100	100	Product
Bottle	3	6	3	20	0	20	0	27	73	27	73	73	Container, Vessel
Bowl	3	9	3	27	0	13	0	20	0	24	0	0	Stadium
Broccoli	17	16	17	60	80	69	80	80	0	100	0	0	Food, Produce, Solid
Cake	7	7	7	33	0	16	0	20	0	100	0	0	Patty, Food, Dish
Carrot	0	10	0	33	53	40	53	53	0	100	0	0	Plant organ, Plant part
Chair	17	8	17	40	0	14	0	20	33	28	33	33	Instrument
Clock	0	41	0	100	0	13	0	20	0	28	0	0	Instrument
Couch	0	19	0	70	0	13	0	20	0	32	0	0	Coloring material, Covering
Cup	13	5	13	37	0	14	0	20	20	21	20	20	Food, Drug, Agent, Fluid
Donut	7	4	7	30	0	21	0	33	0	100	0	0	Food, Doughnut, Solid
Fork	0	8	0	37	0	14	0	20	0	36	0	0	Article, Cutlery
Glass	10	4	10	30	0	16	0	20	0	27	0	0	Methamphetamine, Drug, Agent
Knife	0	8	0	13	0	15	0	20	0	40	0	0	Instrument
Laptop	0	10	0	30	0	22	0	40	33	33	33	33	Machine
Microwave	0	31	0	87	0	34	0	40	0	55	0	0	Commodity, (Home, Kitchen) appliance
Orange	0	7	0	27	0	33	0	33	0	100	0	0	Coloring material
Oven	77	32	77	100	0	40	0	53	0	61	0	0	Commodity, (Home, Kitchen)
Phone	0	9	0	23	0	16	0	27	0	34	0	0	Language unit
Pizza	3	14	3	40	0	24	0	33	0	100	0	0	Food, Dish
Plant	23	8	23	37	0	14	0	20	0	40	0	0	Building complex
Refrigerato	0	39	0	100	33	34	33	40	0	50	0	0	Commodity, Home appliance
Sandwich	30	11	30	37	13	22	13	33	0	100	0	0	Food, Dish
Sink	30	27	30	100	0	34	0	40	100	50	100	100	Container, Vessel, Cesspool, Excavation
Spoon	13	10	13	20	0	10	0	13	67	33	67	67	Container, Article, Cutlery
Table	20	13	20	40	7	16	7	27	27	22	27	27	Food, Board
Toaster	0	11	0	43	0	37	0	47	0	52	0	0	Commodity, (Home, Kitchen) appliance
Tv	0	25	0	100	0	29	0	40	0	32	0	0	Television, Medium
Vase	17	19	17	30	0	14	0	20	40	31	40	40	Container, Vessel

Note: Darker shading equates to higher scores. top three location, top one color property, top one material property

Figure 10: Accuracy of all class labels for location, color, material, and type.

cup is blue does not guarantee the next is blue. For material, the LKB performs well at predicting material as it captures that most bottles in the environment are plastic.

However, it is when we look at the color, that we gain interesting insight about object properties. We see that the AKB follows a slightly different trend than we observed in the object location experiment. We expected that with highly varying properties that the AKB could provide more insight than the LKB. However, if we look deeper at the results of the AKB, we discover that for the class bottle, the AKB has no prediction for color. We believe this points to an important distinction between the variability of an object property and the variability of an object location. When an object's property can take on almost any value (e.g., bottles can be pretty much any color), general knowledge offers little to no insight as to what property the object might have. Furthermore, this situation is also difficult for LKB to learn as the best we can hope for is chance. This suggests that for certain object properties, the only approach to predicting object properties that are highly variable is to remember the exact properties of the instance or perform directly reasoning using lower level features of the object. For both location and properties, we variability effects various accuracy levels of the AKB and LKB. To fully understand the extent in which this insight can be extended to a larger number of classes and properties, we perform a simulated experiment that looks at the variance accuracies across all described classes.

Simulated Evaluation

We exhaustively evaluate how different sources of information impact the various queries listed in *Abstract Knowledge Base* using a similar procedure and experimental setup for each query to *Building the Knowledge Bases*. Specifically, we populated a simulated world of object instances, and randomly assigned attribute values (seen in Fig. 3) and locations (seen in Fig. 6). Properties and locations were made class specific to better capture the real-world (e.g., no couch instances could be located in a drawer and televisions cannot be made of paper). While the rules set in simulation may not capture the rules of a specific real-world environment, they do capture the relationship between class variability and LKB accuracy and can be viewed as a unique layout of a specific home.

Evaluation Metric and Results

To test each query type, we start with a set of simulated instances. This set is taken as the true state of the world. Then a set of world state observations are created by randomly selecting locations and properties for each instance in the world and repeating the process for the number of world state observations. This set of world state observations were used as actual data for the LKB to process and store. To validate our hypothesis in *Experiments*, the evaluation was done similarly to that of the robot experiment where we report the top three locations and top one property. For the last query, object types (*IsA*), was tested by comparing the results of the returned values to three sets of human generated labels base on common sense for the home environment (e.g., *IsA*(Apple, Fruit) is true whereas *IsA*(Bowl, Stadium) is false).

In *Experiments*, we see a limited view of object locations and classes. By doing the simulated evaluation, we can look at if the trends seen in the robot experiment were reflected in the 31 different class types. The results of this evaluation are in Fig. 10. The table shows the accuracy of the AKB and LKB for location, color, and material by class. They are further broken down into accuracy values after seeing one observation vs seeing all 15 observations. The table also includes the different *IsA* relations for each object class.

We can see that several of the trends observed in the robot experiment hold true. For example, ovens, which are less variable in location, have a higher initial AKB accuracy than the LKB. The LKB learns the oven location perfectly after 15 observations. In general, color, which varies highly does poorly for both ABK and LKB unless the object has a notion of a color (e.g., carrot and broccoli). We see that the AKB does well on the material property if the class has a typical material it is made out of (e.g., books, sink, spoon). We test this on an aggregate scale in the next section. For the *IsA* queries, the average accuracy of the relations was 72%. Between the three sets of human labels, there was an 83.17% average pairwise percent agreement.The accuracy values between all three users were within 2% of each other. We can look at Fig. 10 to see that this accuracy can be reflected in the labels produced. It correctly identifies useful types such as apple is a food and bottle is a container. The few cases where *IsA* does not perform well can be seen with bowl being related to stadium and glass to drug.

Role of Variability

The results show that taking into account variability of local knowledge history will be essential for reasoning about new situations. The general trend is that as variability increases, a discount factor should be used to emphasize sources of general knowledge that are resistant to such effects. Fig. 11 was generated by categorizing each simulation output seen in Fig. 10 as either low (1-3 alternatives), medium (4-6 alternatives), or high (7+ alternatives) variability and averaging

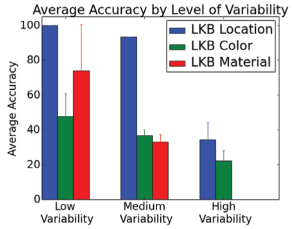

Figure 11: Relationship between LKB accuracy and variability.

all the results for each category. It shows that as variability increases, the LKB accuracy drops. For extreme levels of variability similar to in *Object Properties*, even a general knowledge systems fails. In these situations, a robot should rely on direct observations or chance.

Conclusions

In this work we introduce the SiRoK framework and systematically evaluate it through robot experiments and simulation. We use SiRoK to better understand the trade offs between general knowledge bases that store symbols and concepts and local knowledge bases that store perceptual data. We find that variability is a key heuristic to take into account when evaluating knowledge. In future works, we hope to find methods of fusing the disparate knowledge sources, improving the quality of the BLN in our AKB, and utilizing the *IsA* query.

References

Buschka, P., and Saffiotti, A. 2004. Some notes on the use of hybrid maps for mobile robots. In *Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems*, 547–556.

Chen, J., and Liu, J. 2011. Combining ConceptNet and WordNet for Word Sense Disambiguation. In *International Joint Conference on Natural Language Processing*, 686–694.

Chu, V.; Fitzgerald, T.; and Thomaz, A. L. 2016. Learning object affordances by leveraging the combination of human-guidance and self-exploration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 221–228.

Fung, R., and Chang, K.-C. 2013. Weighing and Integrating Evidence for Stochastic Simulation in Bayesian Networks.

Gabrilovich, E., and Markovitch, S. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In Veloso, M. M., ed., *International Joint Conference on Artificial Intelligence*, 1606–1611.

Garrison, H., and Chernova, S. 2016. Situated structure learning of a bayesian logic network for commonsense reasoning. *CoRR* abs/1607.00428.

Hermans, T.; Rehg, J.; and Bobick, A. 2011. Affordance prediction via learned object attributes. In *International Conference on Robotics and Automation: Workshop on Semantic Perception, Mapping, and Exploration*.

Jain, D.; Kirchlechner, B.; and Beetz, M. 2007. Extending markov logic to model probability distributions in relational domains. In *KI 2007: Advances in Artificial Intelligence*. Springer. 129–143.

Jain, D.; Waldherr, S.; and Beetz, M. 2009. Bayesian Logic Networks. Technical report, Technische Universität München, München.

Lenat, D. B. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

Lin, T.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: common objects in context. *CoRR* abs/1405.0312.

Matuszek, C.; Cabral, J.; Witbrock, M. J.; and DeOliveira, J. 2006. An introduction to the syntax and content of cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 44–49. Citeseer.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Richardson, M., and Domingos, P. 2006a. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Richardson, M., and Domingos, P. 2006b. Markov logic networks. *Machine learning* 62(1-2):107–136.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23(3):309–314.

Saxena, A.; Jain, A.; Sener, O.; Jami, A.; Misra, D. K.; and Koppula, H. S. 2014. Robobrain: Large-scale knowledge engine for robots. *arXiv preprint arXiv:1412.0691*.

Sinapov, J.; Schenck, C.; Staley, K.; Sukhoy, V.; and Stoytchev, A. 2014. Grounding semantic categories in behavioral interactions: Experiments with 100 objects. *Robotics and Autonomous Systems* 62(5):632 – 645. Special Issue Semantic Perception, Mapping and Exploration.

Speer, R., and Havasi, C. 2012. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*.

Stoica, E., and Hearst, M. A. 2004. Nearly-Automated Metadata Hierarchy Creation. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 117–120.

Sun, Y.; Bo, L.; and Fox, D. 2013. Attribute based object identification. In *2013 IEEE International Conference on Robotics and Automation*, 2096–2103.

Tenorth, M., and Beetz, M. 2009. Knowrobknowledge processing for autonomous personal robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 4261–4266. IEEE.

Tsatsaronis, G.; Varlamis, I.; and Vazirgiannis, M. 2008. Word Sense Disambiguation with Semantic Networks. In Sojka, P.; Horák, A.; Kopeček, I.; and Pala, K., eds., *Text, Speech, and Dialogue*, 219–226. Springer.

Waibel, M.; Beetz, M.; Civera, J.; d'Andrea, R.; Elfring, J.; Galvez-Lopez, D.; Häussermann, K.; Janssen, R.; Montiel, J.; Perzylo, A.; et al. 2011. Roboearth. *IEEE Robotics & Automation Magazine* 18(2):69–82.

Zhu, Y.; Fathi, A.; and Fei-Fei, L. 2014a. Reasoning About Object Affordances in a Knowledge Base Representation. In *European Conference on Computer Vision*.

Zhu, Y.; Fathi, A.; and Fei-Fei, L. 2014b. Reasoning about object affordances in a knowledge base representation. In *European conference on computer vision*, 408–424. Springer.