Putting People's Common Sense into Knowledge Bases of Household Robots

Lars Kunze, Moritz Tenorth, and Michael Beetz

Intelligent Autonomous Systems Group
Department of Informatics
Technische Universität München
{kunzel,tenorth,beetz}@in.tum.de

Abstract. Unlike people, household robots cannot rely on commonsense knowledge when accomplishing everyday tasks. We believe that this is one of the reasons why they perform poorly in comparison to humans. By integrating extensive collections of commonsense knowledge into mobile robot's knowledge bases, the work proposed in this paper enables robots to flexibly infer control decisions under changing environmental conditions. We present a system that converts commonsense knowledge from the large Open Mind Indoor Common Sense database from natural language into a Description Logic representation that allows for automated reasoning and for relating it to other sources of knowledge.

1 Introduction

Household robots are expected to accomplish an open-ended set of everyday tasks. Hence, they need to understand under-specified commands given by humans and find out on their own what to do, what to look out for, where to search for objects and so on. Many of these decisions depend on the context at hand and are difficult to foresee when writing the robot control program. In addition, the sheer number of possible situations makes hard-coding them practically impossible. Human-like commonsense knowledge would give robots the ability to *infer* the right decisions and become more flexible. This raises the questions how to acquire the commonsense knowledge that humans possess, how to represent it, and how to enable robots to make inferences based on that knowledge.

The approach we propose in this paper is to use existing collections of commonsense knowledge, such as the Open Mind Indoor Common Sense database (OMICS, [1]), and convert them into first-order representations the robot can use. The OMICS project collects commonsense knowledge described in natural language from Internet users. This information is contributed by users by completing template sentences in web forms. For example, a template sentence like "When a potted plant is dry, then the _____ becomes _____." is used for capturing information about causal relationships. These fragments cover different areas, like the objects found in different rooms, the correct action to take in a situation, or possible problems that can occur while performing a task. For most humans, such information appears trivial, and they can intuitively answer such questions. For a robot, however, this gives valuable information that can

hardly be obtained otherwise. The information entered in the web forms is reviewed by the OMICS project team, post-processed and stored in a relational database.

However, this information cannot directly be used by robots: It is written in colloquial English language, which makes it hard for a robot to interpret it (for example, to find out that "turn on", "switch on" and "start" actually mean the same thing), to relate it to other sources of information like semantic environment maps, and to perform automated reasoning. Therefore, our system first transforms the statements into a formal representation in Description Logic by resolving the meaning of words and mapping them to ontological concepts. This allows to use well-established reasoning techniques, and further enables the robot to integrate the commonsense knowledge into its knowledge processing system [2].

The rest of the paper is structured as follows: We start with an overview of related work (Section 2) and a description of the system architecture (Section 3), explain the extraction and formalization of knowledge from OMICS (Section 4) and elaborate on the integration into the robot's knowledge base (Section 5). Section 6 gives examples of how the robot uses the knowledge. Finally, Section 7 concludes this paper by discussing some of the problems encountered and some remaining challenges for further research.

2 Related Work

Equipping computers (or robots) with common sense is not a new endeavor in Artificial Intelligence: Early work has been done by McCarthy already in 1959 [3]. However, the problem is still far from being solved. As Minsky [4] pointed out, computers require large amounts of commonsense knowledge in order to become concerned with human affairs. Furthermore, he points to the problem that much of the commonsense knowledge has never been described because its information always seemed so obvious. This problem is addressed within the knowledge capturing projects Open Mind Common Sense [5] and Open Mind Indoor Common Sense [1] that acquire commonsense knowledge from web users. Several approaches have been proposed to make use of this information (ConceptNet [6], LifeNet [7], and [8]) for textual-information management, modeling of human activities, or robot task planning. Gupta and Pedro [9] use Bayesian networks to infer the most likely response in a given situation based on knowledge from OMICS. Pentney et al. [10] propose to use the knowledge, represented in a probabilistic model, for tracking the state of the world on the basis of sensor data. However, none of these approaches has tackled the problem of converting the knowledge from natural language to well-defined ontological concepts, which is required for automated reasoning.

3 System Overview

Figure 1 illustrates the process of extracting, formalizing, and reasoning about commonsense knowledge from the OMICS project. First, the system applies natural language processing techniques like part-of-speech tagging and syntax parsing to the knowledge. Then, it resolves the word's meanings to cognitive synonyms (synsets) in WordNet, and exploit mappings between these synsets and concepts in OpenCyc, which already exist for thousands of concepts. Based on OpenCyc's concept definitions, it generates

a formal representation describing the OMICS database relations in Description Logic which becomes part of the robot's knowledge base and can be queried via a Prolog-based interface.

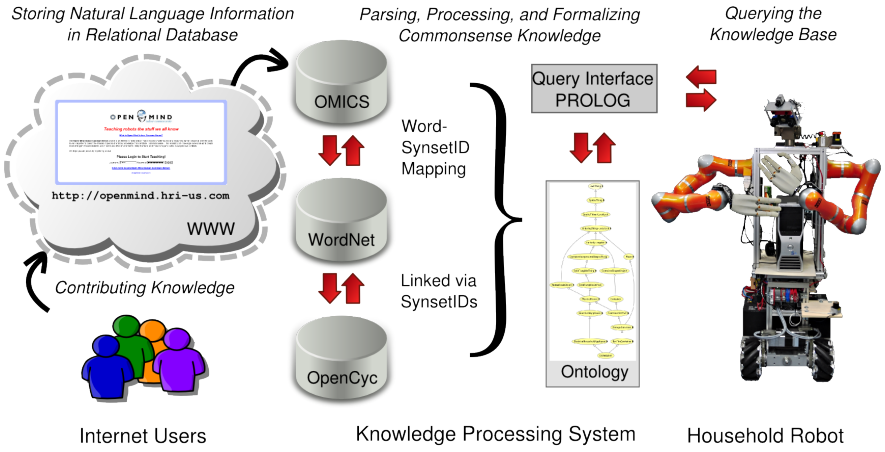

Fig. 1. Overview of the proposed system. Knowledge that is contributed by Internet users is transformed from natural language into a logical representation and can be queried by the robot control program.

4 Formalizing the Knowledge in OMICS

This section describes how the knowledge is converted from the sentences in natural language in OMICS's semi-structured database tables to meaningful first-order representations.

4.1 Natural Language Processing

In our system, the knowledge is represented as object- or action-centered first-order representations, which we need to extract from the OMICS database. For simple relations, like *locations(Object,Room)*, the database columns already provide this information and describe objects, actions, or properties. More complex relations, however, like *problems(Task,Problem)*, are described by short natural language phrases. Figure 2 (left) lists some examples. These phrases need to be interpreted in order to extract information about objects, actions, and properties. First, each word is tagged with its syntactic category, and phrases like a verb phrase (VP) or a noun phrase (NP) are identified using the Stanford Parser [11], a probabilistic context-free grammar parser. The part-of-speech tags (PoS) describe the role (or syntactic category) an expression has in a sentence. Figure 2 (right) shows the parse trees and the PoS annotations[1] of the words in the command "load the dishwasher" and the related problem "dishwasher is full".

Having assigned the PoS tags, the system knows if a word describes an object (usually a noun), an action (in commands normally verbs), or a property (an adjective). This

[1] DT - Determiner; NN - Noun, singular or mass; VB - Verb, base form; VBZ - Verb, 3rd person singular present; JJ - Adjective

Task	Problem
load the dishwasher	dishwasher is full
load the dishwasher	cannot open dishwasher
unload the dishwasher	dishes not dry
unload the dishwasher	dishes are dirty

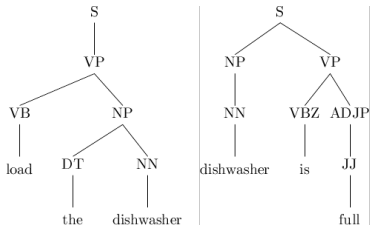

Fig. 2. Left: Examples of tasks and their respective problems taken from the OMICS database. Right: Parse trees of the first task-problem tuple. The part-of-speech information is utilized for extracting object-/action-centered information from the natural language descriptions.

information is used to further interpret the relations in the database. For instance, in the relation *problems(Task,Problem)*, a task is described by an action and an object, whereas problem specifications consist of an object and a property. The relation can thus be written as

$$problems(\overbrace{Object1, Action}^{Task}, \overbrace{Object2, Property}^{Problem}).$$

and the first example in Figure 2 (left) becomes

$$problems(\overbrace{dishwasher, load}^{load\ the\ dishwasher}, \overbrace{dishwasher, full}^{dishwasher\ is\ full}).$$

In this form, the *problems* relation only comprises actions, objects, and properties and can thus be represented in our knowledge processing system.

4.2 Word sense resolution

To make the extracted information usable for automated reasoning, the words need to be transformed from (ambiguous) natural language to well-defined ontological concepts. This transformation is done using the lexical database WordNet, in which words are grouped into sets of cognitive synonyms (synsets). For thousands of these synsets, there are mappings to the OpenCyc ontology, which are used to transform words to ontological concepts.

During this mapping process, the system resolves ambiguities caused by different words that mean the same thing. For example, the expressions “turn on” and “switch on” are both mapped to concept *TurningOnPoweredDevice*. Table 1 shows some examples of synonymous words that are mapped to the same concept. Such a mapping is very important to make use of the full knowledge in OMICS: Without this resolution of the words’ meanings, the system could not detect that e.g. statements about someone mopping or wiping the floor can be combined. Since the knowledge was acquired by many untrained users, the variation in words used is rather high, making the word sense resolution even more important.

A second kind of ambiguity is caused by different meanings of a word: For example, the word “dishwasher” can denote a household appliance or a person cleaning the dishes. Technically, this means to select the right concept the word is to be mapped to.

Type	OMICS	Synset-ID	Cyc Name
Action	turn on	V01468130	*TurningOnPoweredDevice*
	switch on	V01468130	*TurningOnPoweredDevice*
	clean	V01490246	*Cleaning*
	mop	V01352869	*Cleaning*
	sanitize	V01207357	*Cleaning*
	steam clean	V01207630	*Cleaning*
	wipe up	V01352869	*Cleaning*
Property	on	A01599324	*DeviceRunning*
	turned on	A02066470	*DeviceRunning*
	dirty	A00394641	*Dirty*
	unclean	A00394641	*Dirty*
	soiled	A00394641	*Dirty*

Table 1. Mappings for different word types.

The system uses different sources of information for this task: The parsing and PoS tagging determined the type of a word, so that the algorithm can determine if e.g. "clean" is used as a verb (denoting an action) or as an adjective (describing a property).

If several meanings of a word fall within a single syntactic category, the ambiguities are harder to resolve, for example for the word "dishwasher". However, we know from OMICS that the word denotes an object, and can use the Cyc ontology to select the household appliance and discard the second meaning since a person is not an object. In general, we constrain the resulting ontological concepts of objects and actions to be subclasses of *PartiallyTangible* and *ActionOnObject* respectively. Furthermore, rooms in the *locations(Object,Room)* relation need to be a subclass of *RoomInAConstruction*.

Finally, the following numbers should give the reader a basic idea about word sense resolution. In total, the *locations(Object,Room)* table holds 416 distinct entries for objects that are typically found in a kitchen. From these, we could automatically map 247 object descriptions to 178 distinct ontological concepts. The 169 entries that could not be resolved directly were not found in the WordNet search. At least 18 of these unmapped entries could be mapped after truncating the ending *'s'*, i.e. making plural expressions singular. By looking at the remaining objects most of them are described by two or more words, e.g. "milk bottle", "tea pitcher" or "box of donuts". For improving the performance of our system we will consider more sophisticated techniques like stemming and compound processing.

5 Integration into the Robot's Knowledge Base

For our robot to use the knowledge, we need to represent the relations inside KnowRob [2], its knowledge processing system. KnowRob uses OWL-DL [12], a language based on Description logic (DL) [13], to represent the robot's knowledge. Concepts and relations in KnowRob are the same as in OpenCyc, so that the mappings determined in the previous step can be used here. These concepts are now related according to relations in OMICS and added to the knowledge base. For example, the simple relation *parts(dishwasher,motor)* is transformed to

$$\begin{aligned} &parts(dishwasher, motor) \\ &\quad \Rightarrow ((wordnetCyc(dishwasher, Dishwasher) \wedge wordnetCyc(motor, Engine)) \\ &\qquad \Rightarrow properPhysicalParts(Dishwasher, Engine)) \end{aligned}$$

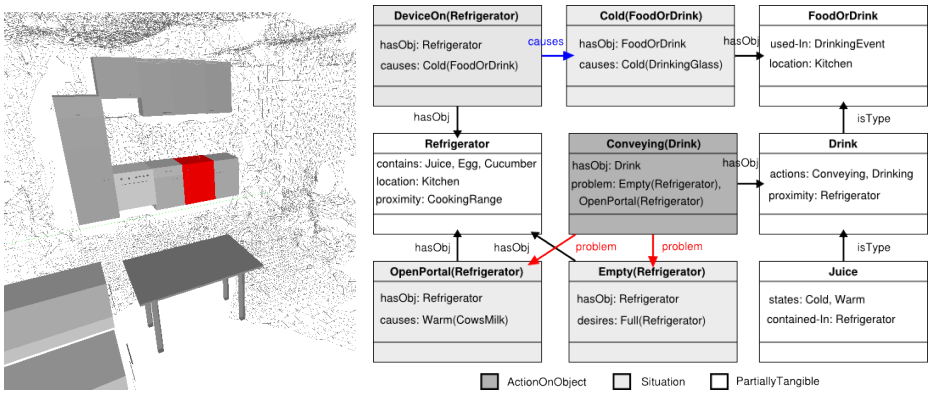

Fig. 3. Left: Refrigerator located in the integrated semantic environment map. Right: Simplified view on the ontology that was generated from the OMICS database.

where the words "dishwasher" and "motor" are mapped to the concepts *Dishwasher* and *Engine* respectively. These concepts are related by the role *properPhysicalParts*. A more complex example of a causal relationship is given by

$$
\begin{aligned}
&causes(dishwasher, used, cup, clean)\\
&\Rightarrow ((wordnetCyc(dishwasher, Dishwasher) \wedge wordnetCyc(used, UsedArtifact) \wedge\\
&\quad wordnetCyc(cup, DrinkingMug) \wedge wordnetCyc(clean, Clean))\\
&\quad \Rightarrow \textit{causes-SitSit}(\\
&\quad\ SituationInWhich(Dishwasher, UsedArtifact),\\
&\quad\ SituationInWhich(DrinkingMug, Clean)))
\end{aligned}
$$

where the resulting concepts are related by the role *causes-SitSit*. Since roles in DL can only represent relations that have an arity of exactly two, we represent object-property tuples as sub-concepts of type *Situation* and object-action tuples as sub-concepts of type *ActionOnObject*.

6 Applications

By integrating the OMICS knowledge into an ontology, we can perform reasoning about similar objects and/or sub-classes of objects. Additionally, KnowRob provides methods for accessing sensor data, for instance an semantic environment model and objects detected by the robot's vision system. These instances can be related to the class knowledge obtained from OMICS for determining in which room one can typically find an object, what are possible object states, or what are potential problems when applying a certain action to an object.

The following scenario illustrates what kind of knowledge from the OMICS database can be used by a household robot. We consider a situation where the robot is supposed to serve a cold drink to a person. To accomplish the task the robot has to infer where in its environment it can find a drink that is cold. Some of the relevant knowledge that is described in the text and is referred in the example queries is depicted in Figure 3.

The first task of the robot is to resolve the meanings of "cold" and "drink", which are *Cold* and *Drink* respectively. Then it has to identify causes that have the effect of a cold drink. In the robot's knowledge base there exist no causal rule which explicitly

involves a situation of type *Cold(Drink)*, but however, there exist a more general rule that involves the concept *Cold(FoodOrDrink)*. Since the robot knows from its ontology that *Drink* is a specialization (or sub-class) of *FoodOrDrink* it can make use of more general rule, which reads as follows:

$$DeviceOn(Refrigerator) \Rightarrow_{causes} Cold(FoodOrDrink).$$

From the above rule the robot infers that an instance of type *Refrigerator* with property *DeviceOn*, can cause cold drinks, which is shown by the following query:

```
?- subClassOf('Drink', SuperCls),
   situation(Sit), hasObj(Sit, SuperCls), hasProperty(Sit,'Cold'),
   owl_restriction_on(Cause, restriction(causes,some(Sit))).

   SuperCls  = 'FoodOrDrink';
   Sit       = 'Cold(FoodOrDrink)',
   Cause     = 'DeviceOn(Refrigerator)'.
```

Furthermore, the robot knows from OMICS's (spatial-) relationships that things like *Egg-Foodstuff, Cucumber-Foodstuff,* and *Juice* are typically contained (*in-ContGeneric*) in a *Refrigerator*. Since *Drink* is a super-class of *Juice*, the robot deduces that it could get an instance of type *Juice* from the fridge which, regarding the causal rule, should be cold. From OMICS' *locations* relation the robot knows that a *Refrigerator* is typically found in a *Kitchen*. By accessing the semantic environment map through KnowRob, the robot can precisely locate an instance of type *Refrigerator*. The result of the following query is depicted in Figure 3 (left).

```
?- owl_individual_of(Fridge, 'Refrigerator').

   Fridge = fridge1.
```

However, in the case that the fridge has not been mapped yet, the robot can use OMICS's *proximity* relation and try to find the fridge by localizing objects near-by, e.g. instances of type *CookingRange* or *Freezer*. Finally having located the fridge in the kitchen, the robot can verify with its vision system whether there is some juice inside. While getting the cold drink from the fridge the robot can watch out for potential problems that are related to the *Conveying(Drink)* action, e.g. *Empty(Refrigerator)* or *OpenPortal(Refrigerator)*, which are retrieved by the query:

```
?- actionOnObject(Action,'Drink'),
   owl_restriction_on(Action, restriction(problem,some(Problem))).

   Action  = 'Conveying(Drink)',
   Problem = 'Empty(Refrigerator)' ;
   Action  = 'Conveying(Drink)',
   Problem = 'OpenPortal(Refrigerator)'.
```

Although the overall reasoning process in this scenario does not work fully automatically, the emerging sub-queries can be answered based on the knowledge retrieved from OMICS. Furthermore, the scenario should point out the importance and omnipresence of commonsense reasoning within everyday tasks.

7 Discussion and Conclusions

In this paper, we described how commonsense knowledge that is acquired from web users and represented in natural language is transformed into first-order representations which enable robots to reason about everyday tasks. We showed how we extract object-

and action-centered information from natural language phrases stored in the OMICS database, map it to ontological concepts, integrate it in the knowledge processing system KnowRob, and use it for answering task-relevant queries.

Though the processes described in this work are fully automated, some remaining flaws are best resolved by manual revision: First, the mapping between OMICS and OpenCyc is not complete, meaning that some words or expressions cannot be transformed to ontological concepts. And second, our rather simple word sense disambiguation strategies may fail to select the correct meaning, but this problem is beyond the scope of this work. Regarding the knowledge provided by OMICS, it should be noted that the information, as it was entered from ordinary Internet users, is first not complete, second redundant, and third even sometimes contradictory. By the techniques presented in this paper, we are able to resolve some ambiguities and to assemble all assertions for a topic by linking the words to concepts — otherwise, the robot would see pieces of information using different words as not related. However, completely resolving these issues remains an open challenge.

8 Acknowledgments

We would like to thank the Honda Research Institute USA Inc. for providing the OMICS data. This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems (CoTeSys)*, see also `www.cotesys.org`.

References

1. Gupta, R., Kochenderfer, M.J.: Common Sense Data Acquisition for Indoor Mobile Robots. In: AAAI. (2004) 605–610
2. Tenorth, M., Beetz, M.: KnowRob — Knowledge Processing for Autonomous Personal Robots. In: IEEE/RSJ International Conference on Intelligent RObots and Systems. (2009)
3. McCarthy, J.: Programs with Common Sense. In: Proceedings of the Teddington Conference on the Mechanization of Thought Processes, London, Her Majesty's Stationary Office (1959)
4. Minsky, M.: Commonsense-based Interfaces. Commun. ACM **43** (2000) 66–73
5. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open mind common sense: Knowledge acquisition from the general public. In: CoopIS/DOA/ODBASE. (2002)
6. Liu, H., Singh, P.: ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal **22** (2004) 211–226
7. Singh, P., Williams, W.: LifeNet: A Propositional Model of Ordinary Human Activity. In: Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP). (2003)
8. Shah, C., Gupta, R.: Building Plans for Household Tasks from Distributed Knowledge. In: Workshop on Modeling Natural Action Selection at IJCAI 2005. (2005)
9. Gupta, R., Pedro, V.C.: Knowledge Representation and Bayesian Inference for Response to Situations. In: AAAI 2005 Workshop on Link Analysis. (2005)
10. Pentney, W., Popescu, A.M., Wang, S., Kautz, H.A., Philipose, M.: Sensor-Based Understanding of Daily Life via Large-Scale Use of Common Sense. In: AAAI. (2006)
11. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA (2003)
12. W3C: OWL Web Ontology Language. `http://www.w3.org/TR/owl-ref/` (2004)
13. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)