Handling Open Knowledge for Service Robots

Xiaoping Chen, Jianmin Ji, Zhiqiang Sui, and **Jiongkun Xie**
Computer School, University of Science and Technology of China
{*xpchen, jianmin*}@*ustc.edu.cn*, {*zqsui, devilxjk*}@*mail.ustc.edu.cn*

Abstract

Users may ask a service robot to accomplish various tasks so that the designer of the robot cannot program each of the tasks beforehand. As more and more open-source knowledge resources become available, it is worthwhile trying to make use of open-source knowledge resources for service robots. The challenge lies in the autonomous identification, acquisition and utilization of missing knowledge about a user task at hand. In this paper, the core problem is formalized and the complexity results of the main reasoning issues are provided. A mechanism for task planning with open-knowledge rules which are provided by non-experts in semi-structured natural language and thus generally underspecified are introduced. Techniques for translating the semi-structured knowledge from a large open-source knowledge base are also presented. Experiments showed a remarkable improvement of the system performance on a test set consisting of hundreds of user desires from the open-source knowledge base.

1 Introduction

Users may ask a service robot to accomplish various tasks in various contexts. This extensive diversity of tasks and contexts makes it infeasible for the designer of the robot to predict and program all of the possible tasks beforehand. Autonomous task planning techniques for service robots have been developed to overcome this difficulty [Burgard *et al.*, 1999; Keller *et al.*, 2010]. However, frequently there are gaps between a user task and a robot's knowledge/skills, which make it impossible for the robot's planner to work out a plan for the task. This in turn leads to the notorious bottleneck problem of knowledge acquisition. For instance, when a user tells a robot, "I am hungry," the robot can make a plan to meet the user's desire only if the robot knows some means to the end, say, by serving food. Similarly, knowledge plays an important role for other purposes (such as human-robot interaction [Kruijff *et al.*, 2010; Lemaignan *et al.*, 2012a]). As more and more open-source knowledge resources become available (e.g., large-scale knowledge bases, ontologies, household appliance manuals, and through human-robot communication), there are opportunities to make use of open knowledge, i.e., the knowledge from these open-source knowledge resources for robot task planning and resolve the difficulty of knowledge acquisition.

In some sense, one can take the entire body of open knowledge as the knowledge base of a robot/agent. However, different tasks generally need different bodies of knowledge and planning with the union of all open-source knowledge resources suffers from hard problems of inconsistency and inefficiency. Furthermore, a great proportion of open knowledge is expressed in natural languages and it is impossible to require that the knowledge is formalized. We proposed an approach to utilizing open knowledge [Chen *et al.*, 2012; Xie *et al.*, 2012] in which the challenge lies in the autonomous identification, acquisition and utilization of missing knowledge about a user task at hand. In this paper, a formal basis for the identification of missing knowledge is put forth, providing useful insights for the design of robots that can autonomously gain and utilize open knowledge. Techniques for translating the semi-structured knowledge from a large open-source knowledge base are developed and a mechanism for planning with the underspecified open-knowledge rules is also introduced. There are previous efforts that share common concerns with us about open knowledge [Cantrell *et al.*, 2012; Chen *et al.*, 2010; Fong *et al.*, 2003; Lemaignan *et al.*, 2012b; Mohan *et al.*, 2012; Rosenthal *et al.*, 2010; Schiffer *et al.*, 2012; Talamadupula *et al.*, 2010]. To our knowledge, there has been little work on the specific research issues we report here.

Section 2 explains the motivations behind this effort, including the technical challenges and the main ideas of this work. Section 3 presents the formalization of knowledge gaps/rehabilitations and the complexity results. Section 4 describes the techniques to translate semi-structured open knowledge from a large open-source knowledge base into an internal representation. Section 5 reports the mechanism of planning with the generated, underspecified rules and the experimental results. Section 6 draws conclusions.

2 Motivations

We assume that a service robot is equipped with an action model, i.e., a set of primitive actions and each of them can be executed by the robot through running of a corresponding low-level routine. Each primitive action is described with a pair $\langle pre(a), \mathit{eff}(a)\rangle$, where $pre(a)$ and $\mathit{eff}(a)$ are the preconditions and effects of a, respectively. The robot's local knowledge may include other background knowledge besides the action model. To concentrate on our main goals, in this paper we only consider the situations where a robot can accomplish a user task if only it gains more knowledge and focus on the following technical challenges.

(1) **Knowledge gaps:** Given a user task t to the robot with local knowledge LK, there may exist knowledge gaps between t and LK [Chen *et al.*, 2012], which prevents the robot from generating any plan for the task. A fundamental issue of this research is to characterise knowledge gaps between t and LK, establishing a formal basis for searching and utilizing the missing knowledge from open-source knowledge resources to plan for and fulfill the user task.

(2) **Translation:** It is a long dream of and a great challenge to AI to make a machine understand natural languages. In this paper, we aim at a much smaller goal—to translate open knowledge in semi-structured English into an internal representation that can be made use of in robot planning. The Open Mind Indoor Common Sense (OMICS) database [Gupta and Kochenderfer, 2004] is employed as the main source of open knowledge investigated in this paper.

(3) **Underspecification:** Open knowledge is correct in human understanding, but generally underspecified [Chen *et al.*, 2010]. Particularly, preconditions and effects of an OMICS task, which corresponds to a (high-level) action, are generally absent. For instance, there is a rule in the *Tasks/Steps* table of OMICS

trash an object ←
get object, find trash can, put object in trash can.

Here *trash an object* can be taken as a high-level action. However, currently it is too difficult to gain the complete preconditions and effects of these actions from open knowledge. Consequently, it is not feasible to apply the classical planning methods to OMICS rules directly.

The main ideas of this paper are sketched as follows. The first and major challenge is formalized as a variant of abduction [Console *et al.*, 1991; Eiter *et al.*, 1997], i.e., the problem of abducting from open-source knowledge resources a set of rules that "rehabilitates" the robot's local knowledge, so that the user task at hand can be planned with the enlarged body of knowledge. A knowledge gap is defined as a minimal rehabilitation, which specifies a least amount of missing knowledge with respect to the user task and the robot's local knowledge. We present complexity results about the relevant reasoning problems, which provide insights about the design of algorithms for open knowledge acquisition: (i) Finding a knowledge gap may not be the most efficient way to rehabilitate the local knowledge of a robot, though superficially this would be the most efficient way. (ii) The expressiveness of the internal representation of open knowledge also impacts on the computational complexity to a great extent. We identify a tractable case where a rehabilitation can be computed in polynomial time.

A set of techniques for translating the semi-structured knowledge in OMICS are developed, with which the semantic representation of OMICS tasks and rules can be produced. A new mechanism for planning with underspecified rules from OMICS is also introduced. In principle, it first decomposes a user task into a sequence of a robot's primitive actions with OMICS rules in a way similar to HTN planning [Erol *et al.*, 1995], whereas preconditions/effects are not necessary in the process. Then it checks the executability of the generated action sequence. This is feasible since the preconditions and effects of primitive actions are included in the robot's local knowledge base and they can be bounded with the semantic information of the OMICS tasks corresponding to the primitive actions.

3 Formulation of Knowledge Gaps

3.1 Background

The language of causal theories [McCain and Turner, 1997] is based on a propositional language with two zero-place logical connectives $\top$ for tautology and $\bot$ for contradiction. We denote by $Atom$ the set of atoms, and Lit the set of literals: $Lit = Atom \cup \{\neg a \mid a \in Atom\}$. Given a literal l, the *complement* of l, denoted by $\bar{l}$, is $\neg a$ if l is a and a if l is $\neg a$, where a is an atom. A set I of literals is called *complete* if for each atom a, exactly one of $\{a, \neg a\}$ is in I. In this paper we identify an interpretation with a complete set of literals. Let I be an interpretation and F a propositional formula, I *satisfies* F or I is a *model* of F, denoted $I \models F$, is defined as usual.

A *causal theory* is a finite set of *casual laws* of the form:

$$\phi \Rightarrow \psi, \qquad (1)$$

where ϕ and ψ are propositional formulas. Intuitively, the causal law reads as "ψ is caused if ϕ is true". A causal law of the form (1) is *definite* if ψ is a literal and ϕ is a conjunction of literals. A causal theory is *definite* if all causal laws in it are definite.

Let T be a causal theory and I an interpretation. The *reduction* T^I of T w.r.t. I is defined as $T^I = \{\, \psi \mid \text{for some } \phi \Rightarrow \psi \in T \text{ and } I \models \phi \,\}$. T^I is a propositional theory. We say that I is a *model* of T if I is the unique model of T^I.

For any causal theory T and a propositional formula F, we say that T *credulous entails* F, denote $T \vdash_c F$, if there exists a model I of T such that $I \models F$. The credulous entailment is non-monotonic in the sense that, after adding other causal laws a propositional formula may no longer be entailed. For example, a causal theory $T = \{p \Rightarrow p\}$, its only model is $\{p\}$ and $T \vdash_c p$. Let $T' = \{p \Rightarrow p, \top \Rightarrow \neg p\}$, its only model is $\{\neg p\}$ and $T' \vdash_c \neg p$.

Compared with Situation Calculus [Reiter, 2001] and other formalisms for reasoning about action based on classical logic [Van Harmelen *et al.*, 2008], causal theories allow for convenient formalization of many challenging phenomena such as the frame problem, indirect effects of actions (ramifications), implied action preconditions, concurrent interacting effects of actions, and things that change by themselves [Van Harmelen *et al.*, 2008]. These features make the causal-theoretical language suitable for formalizing open knowledge which is expressed in natural language and used for service robots. [Cantrell *et al.*, 2012] described an example in a search and rescue scenario in which a robot is searching a building that is unsafe for human exploration. At the beginning of the exploration task, the robot's knowledge specifies that the robot should enter any room it encounters through an open door. During the search operation, however, the robot gains a new piece of knowledge that the building's doors are all designed to unlatch when the fire alarm is triggered. In that case, the robot should push the doors open and search rooms behind them. Using the causal-theoretical language, the robot's local knowledge base can be updated very easily: Simply adding new rules for the newly known context into the robot's local knowledge base, while keeping all old rules unchanged since they are still valid for the previously known contexts.

In this paper, we consider causal theories as the formalism for action domains of service robots. Then open knowledge bases are viewed as sets of causal laws whose elements could be added to the local knowledge base of a robot.

3.2 Computational Complexity

Definition 1 *A* knowledge rehabilitation problem (KRP) *is a triple $\langle A, T, O\rangle$, where A and T are causal theories, and O is a propositional formula.*

Intuitively, A specifies the local knowledge base of a robot, T is an open-knowledge base (assumed as a set of causal laws), and O represents the set of tasks that need to be accomplished. We say KRP $\langle A, T, O\rangle$ is *definite* if A and T are definite causal theories, and O is a conjunction of literals.

Definition 2 *Let $P = \langle A, T, O\rangle$ be a KRP and $E \subseteq T$. E is a* credulous rehabilitation *for P if there exists a model I of $A \cup E$ such that $I \models O$.*

Similarly we can define a skeptical rehabilitation which requires $A \cup E$ has a model and O is satisfied by every model of $A \cup E$. In this paper, we only consider credulous rehabilitation.

Definition 3 *Let $P = \langle A, T, O\rangle$ be a KRP and $E \subseteq T$. E is a* knowledge gap *of A w.r.t. T and O (for P), if E is a credulous rehabilitation for P and any proper subset of E is not a credulous rehabilitation for P.*

Definition 4 *Let $P = \langle A, T, O\rangle$ be a KRP and a causal law $r \in T$. P is* credulous consistent *if there exists a credulous rehabilitation for P. r is* credulous relevant *for P if $r \in E$ for some credulous rehabilitation E for P. r is* credulous necessary *for P if $r \in E$ for every credulous rehabilitation E for P.*

Definition 5 *Let $P = \langle A, T, O\rangle$ be a KRP and a causal law $r \in T$. P is* consistent *if there exists a knowledge gap for P. r is* relevant *for P if $r \in E$ for some knowledge gap E for P. r is* necessary *for P if $r \in E$ for each knowledge gap E for P.*

These concepts can help us to locate missing knowledge for a KRP. Now we consider the computational complexity related to them.

From Proposition 3 and Proposition 6 in [Giunchiglia *et al.*, 2004], we have the following theorem.

Theorem 1 *Let $P = \langle A, T, O\rangle$ be a KRP and $E \subseteq T$. Determining whether E is a credulous rehabilitation for P is Σ_2^P-complete. If P is definite, then the problem is NP-complete.*

Given a KRP $P = \langle A, T, O\rangle$, one can construct a causal theory T_P which contains:

- A,
- $a_r \wedge \phi \Rightarrow \psi$ for each causal law $r \in T$ of the form (1),
- $a_r \Rightarrow a_r$ and $\neg a_r \Rightarrow \neg a_r$ for each $r \in T$,
- $\neg O \Rightarrow \bot$,

where a_r is a new atom for each causal law $r \in T$. Note that, if P is definite, one can equivalently replace $\neg O \Rightarrow \bot$ by the causal laws $\bar{l} \Rightarrow \bot$ for each literal l occurred in O. Then T_P would be a definite causal theory.

Proposition 1 *Let $P = \langle A, T, O\rangle$ be a KRP and $E \subseteq T$. E is a credulous rehabilitation for P iff there exists a model I of T_P such that $E = \{r \in T \mid a_r \in I\}$.*

The computational complexity results are summarized in Table 1. Each entry represents completeness for the corresponding class. The entries in the column under "Arbitrary" are complexity results of arbitrary KRPs and the entries under "Definite" are complexity results of definite KRPs. The entries in the row of "Recognition" are complexity results for the problem of determining whether a set $E \subseteq T$ is in the corresponding class.

KRP P	Credulous Rehabilitation		Knowledge Gap	
Problem:	Arbitrary	Definite	Arbitrary	Definite
Recognition	Σ_2^P	NP	Π_2^P	coNP
Consistency	Σ_2^P	NP	Σ_2^P	NP
Relevance	Σ_2^P	NP	Σ_3^P	Σ_2^P
Necessity	Π_2^P	coNP	Π_2^P	coNP

Table 1: Computational complexity results

4 Translation of Semi-structured Knowledge

In the OMICS project, an extensive collection of common sense knowledge for indoor environments was collected from non-experts over Internet in order to enhance the capabilities of indoor robots for autonomously accomplishing tasks. At this point, there are 48 tables in OMICS representing different types of common sense knowledge. Some of these tables are so related that they have to be merged together by a SQL query to generate a joint table. As an instance, to represent the knowledge about the task *trash an object* and its involved steps, a joint table *Tasks/Steps* is generated as shown in Table 2.

task	*stepnum*	*step*
trash an object	0	get object
trash an object	1	find trash can
trash an object	2	put object in trash can

Table 2: A joint table *Tasks/Steps*

The knowledge in OMICS is semi-structured but is not formalized. Therefore, it is necessary to translate the OMICS knowledge into a formal expression processable by a robot for its planning. We introduce meta-predicates to represent

Meta-predicate	Explanation
$\text{obj}(N, O)$	O is an object named N
$\text{trash}(O)$	A task that dumps object O as a trash
$\text{get}(O_0, O_1)$	A task that gets object O_0 from the location of object O_1
$\text{find}(O_0, O_1)$	A task that finds object O_0 from the location of object O_1
$\text{place}(O_0, O_1)$	A task that places object O_0 at the location of object O_1

Table 3: A part of meta-predicates

the objects and tasks in OMICS domain. Table 3 shows some example meta-predicates. Besides, there are other meta-predicates defined for capturing the relations between tasks. The meta-predicate $\text{task}(\tau, \textit{Task})$ denotes that *Task* is a specific task whose index is τ. For the purpose of expressing the OMICS knowledge about a task and its involved steps, we defined the meta-predicate $\text{dec}(\tau, \textit{Seq})$ in which τ is the index of the task and *Seq* is a ordered list of its steps (in form of $[\tau_0, \ldots, \tau_n]$). Each element τ in the list *Seq* is the index of a task. For instance, the semantic representation of the OMICS rule expressed in Table 2 is

$$\begin{aligned}
&\text{dec}(\tau_0, [\tau_1, \tau_2, \tau_3]), \text{task}(\tau_0, \text{trash}(X_0)), \text{obj}(\textit{object}, X_0),\\
&\text{task}(\tau_1, \text{get}(X_0, Y_0)), \text{obj}(\textit{object}, X_0),\\
&\text{task}(\tau_2, \text{find}(X_1, Y_1)), \text{obj}(\textit{trash_can}, X_1),\\
&\text{task}(\tau_3, \text{place}(X_0, X_1)).
\end{aligned}$$

Our approach to the knowledge translation consists of two steps: i) *syntactic construction*, where the complete syntax of an OMICS rule is constructed; and ii) *semantic parsing*, taking as input the syntax of the rule and producing its semantic representation.

4.1 Syntactic Construction

In our case, the syntax of an OMICS rule (e.g., that in Table 2) is composed of typed dependencies, including the syntax of its elements and the dependent relations between these elements. The syntax of elements in natural language is represented as the Stanford typed dependencies [de Marneffe and Manning, 2008]. In order to capture the relations between elements in each OMICS rule, the Stanford typed dependencies are augmented: New typed dependency are defined regarding to different types of tables in OMICS. Considering the *Tasks/Steps* table as an example, a new typed dependency *step(governor, dependent)* is defined, meaning that the dependent is a step of the governor.

To construct the syntax of OMICS rules, the syntax of elements in a corresponding table are first parsed through a probabilistic syntactic parser, Stanford parser [de Marneffe *et al.*, 2006]. Then their parses are linked together according to the augmented typed dependencies. Figure 1 shows the syntax of the OMICS rule represented in Table 2.

4.2 Semantic Parsing

Our semantic parsing maps the syntax (typed dependencies, denoted $\mathbf{x}$) of an OMICS rule to its semantic representation (denoted $\mathbf{z}$). The typed dependency grammar consists of the

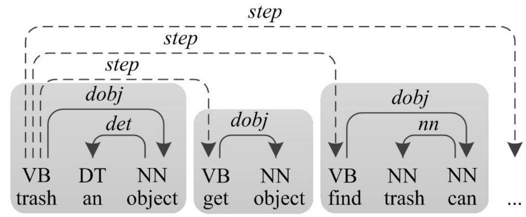

Figure 1: An example of syntax of OMICS rules. The dashed edges marked as *step* are the new typed dependencies defined for the *Tasks/Steps* table.

words with part-of-speech (denoted W) and typed dependent relations (denoted D). In a specific syntax $\mathbf{x}$, the word is denoted as $\mathbf{x}_W$ and the dependency as $\mathbf{x}_D$. Following [Goldwasser and Roth, 2011], our semantic parsing is devided into two stages: *alignment* and *argument connection*. In the alignment stage, a lexical mapping is made indicating that a word $w \in \mathbf{x}_W$ is aligned with a meta-predicate $p \in P$. For example, in Figure 2, a verb *trash* is mapped to a complex meta-predicate $\text{task}(\tau_0, \text{trash}(O_0))$. At the meantime, an OMICS rule is also mapped to a corresponding meta-predicate. In the case of *Tasks/Steps* table, the meta-predicate is $\text{dec}(T, \textit{Seq})$.

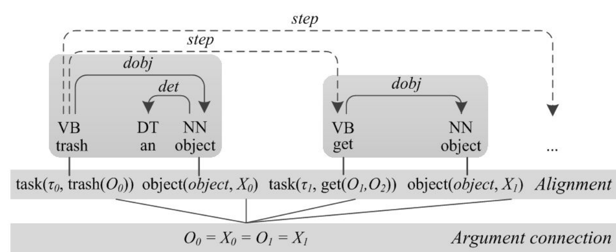

Figure 2: An example of semantic parsing.

The arguments of meta-predicates which are generated in alignment stage are independent. The objective of argument connection is to connect the arguments which are equivalent such as O_0, X_0, O_1, and X_1 in Figure 2. We denote the meta-predicates generated from alignment as $\mathbf{p}$ and the connections of arguments in $\mathbf{p}$ as $\mathbf{z}$. A weighted linear model is employed, as described in [Collins, 2002], to find the most possible $\mathbf{z}$ (i.e., semantic representation) given $\mathbf{x}_D$ and $\mathbf{p}$:

$$\hat{\mathbf{z}} = F_\theta(\mathbf{x}_D, \mathbf{p}) = \arg\max_{\mathbf{z} \in \mathbf{GEN}(\mathbf{p})} \sum_{d \in \mathbf{x}_D} \sum_{z \in \mathbf{z}} \theta \cdot \phi(d, z)$$

where $\phi(d, z)$ is the inner product $\sum_s \theta_s \cdot \phi_s(d, z)$. The function **GEN** enumerates a set of candidates of argument connection given $\mathbf{p}$. θ is a parameter vector over the features. Features are defined as $\phi : D \times Z \to \mathbb{R}^s$ to represent the relations between dependencies in D argument connections in Z. For instance, one of the features might be

$$\phi_3(d, z) = \begin{cases} 1 & \text{if } d \text{ is of type } \textit{dobj} \text{ and} \\ & z \text{ is } \arg(\text{gov}(d), 2) = \arg(\text{dep}(d), 2) \\ 0 & \text{otherwise} \end{cases}$$

where $\arg(p, i)$ returns the ith argument of meta-predicate p,

$\mathrm{gov}(d)$ returns the meta-predicate aligned to the govenor of d, and $\mathrm{dep}(d)$ returns the one aligned to the dependent of d.

5 Planning with Underspecified Rules

5.1 A Tractable Case

Since it is hard to compute a credulous rehabilitation or a knowledge gap even for a definite KRP, here we identify a special sort of KRPs such that a credulous rehabilitation can be computed in polynomial time.

Definition 6 *A KRP $P = \langle A, T, O\rangle$ is called* regular *if there exist sets S and J of atoms such that:*

- *for each atom $a \in S$, a does not occur in A;*
- *each causal law in T is in the form*
$$a_1 \wedge \cdots \wedge a_n \Rightarrow a, \tag{2}$$
where $a \in S$ and $a_1, \ldots, a_n$ belong to $S \cup J$;
- *O is a conjunction of atoms in S;*
- *for any subset $J' \subseteq J$, there exists a model for the causal theory $A \cup \{\neg a \Rightarrow \bot \mid a \in J'\}$.*

Proposition 2 *Let $P = \langle A, T, O\rangle$ be a regular KRP, a credulous rehabilitation for P can be computed in $O(n^2)$ time, where n is the number of causal laws in T.*

Here we sketch the proof and meanwhile briefly explain the intuition behind. Let S and J be the corresponding sets of atoms for P, we can construct a polynomial algorithm for computing a credulous rehabilitation as follows:

1. set the set $U = J$;
2. for each causal law $r \in T$ of the form (2), if $\{a_1, \ldots, a_n\} \subseteq U$ then add the atom a to U and memorize r and all memorized causal laws of a_i $(1 \le i \le n)$ for a;
3. if all atoms in O belong to U, then return the set E of corresponding causal laws for these atoms;
4. if U is not increased then return P is not credulous consistent, else go to 2.

We can prove that E is a credulous rehabilitation for P. Firstly, there exists a model I of $A \cup E$ as for each subset $J' \subseteq J$, $A \vdash_c J'$ and the corresponding atoms in S are also satisfied. Secondly, all atoms in O belong to U, then $A \cup E \vdash_c O$. In the worst case, the iteration would run n times, then the procedure terminates in $O(n^2)$ time.

5.2 Conversion of Semi-structured Knowledge

Section 4 specifies how to translate the knowledge in the *Tasks/Steps* table of OMICS to the meta-predicate representation. For each obtained meta-predicate $\mathrm{act}(\tau, Task)$, we convert it to a causal law, $Task_t \Rightarrow \tau_t$, where $Task_t$ means the task $Task$ is accomplished at time t. For each $\mathrm{dec}(\tau, Seq)$, we convert it to a causal law

$$\tau^1_{t_1} \wedge \cdots \wedge \tau^m_{t_m} \Rightarrow \tau_t,$$

where $\tau^i_{t_i}$ means a task named τ^i is accomplished at time t_i $(1 \le i \le m)$, τ_is are consequently presented in Seq and $t_1 \le \cdots \le t_m \le t$.

We use T to denote the set of causal laws converted from meta-predicate representation of *Tasks/Steps* tables and O to denote the conjunction of task-atoms τ_t for each task τ which needs to be accomplished at time t. Clearly, we can construct a KRP $P = \langle A, T, O\rangle$ such that each credulous rehabilitation for P contains a sequence of actions to accomplish corresponding tasks. Furthermore, if for any set J of action-atoms, there exists a model for $A \cup \{\neg a_t \Rightarrow \bot \mid a_t \in J\}$, then P is regular. Intuitively, it means that the robot could execute any action at any time. Then from Proposition 2, we can compute a credulous rehabilitation for P in polynomial time.

5.3 Algorithms

We developed a set of algorithms of task planning with the translated OMICS rules, which are generally underspecified, for regular KRPs. Algorithm 1 returns an executable action sequence for a given task if possible. Algorithm 2 returns a possible action sequence for *Task*. It first checks whether the task is semantically equivalent to a primitive action of the robot. If not, it computes the set of tasks which are semantically equivalent to the input task in *Task* and tries to plan for one of them which has not been tried before. Algorithm 3 checks whether an action sequence is executable. For this purpose, the algorithm first tries to bound the variables in $pre(a_i)$ and $eff(a_i)$ according to S (i.e., the estimated current state in the planning process) and $sem(as)$, which contains the semantic representations of OMICS tasks corresponding to actions in as. If $pre(a_i)$ or $eff(a_i)$ cannot be bounded completely, it is set *False* and thus makes the action sequence inexecutable. The update of S is conducted by cancelling any element inconsistent with $eff(a_i)$ from S and then adding $eff(a_i)$ to S.

Algorithm 3 does not assume any specific initial state; instead, it only checks if the action sequence itself will not block its execution. It can be proved that Algorithm 3 returns *True* if and only if there is an initial state under which the action sequence can be executed successfully. This executability criterion is rational due to the following reasons. On one hand, it is impossible to collect any initial state in our "simulated" test, though a real robot can collect environmental information through perception. On the other hand, all possible initial states cannot be enumerated and tested one by one.

The algorithms contain more functions than the computation specified in Proposition 2, including computations of semantically equivalent tasks. In our setting, the computation of semantic equivalence is reduced to a limited set of much smaller sub-problems in terms of the meta-predicates, with limited background knowledge mainly from WordNet synonyms. Therefore, the algorithms are still efficient enough.

5.4 Experimental Results

We conducted the experiments with two raw and large test sets collected from OMICS. The first one consists of 11,615 different user tasks from the *Tasks/Steps* table and the second 467 different user desires from *Help*. Five action models, $AM_1 = \{$*move*$\}$, $AM_2 = \{$*move, find*$\}$, $AM_3 = \{$*move, find, pick up*$\}$,

Algorithm 1 getActionSeqeneceForTask(*Task*)

$A := \emptyset$ /* A stores generated action sequence */
repeat
  visitedTask $:= \emptyset$
  $as :=$ generateAS(*Task*, A)
  if $as = \emptyset$ **then return** *False*
  if checkExcutable(as) $=$ *False* **then** $A := A \cup \{as\}$
    else return as
end repeat

Algorithm 2 generateAS(*Task*, A)

if isAction(*Task*) $=$ *True* **then return** $\{$*Task*$\}$
  else *Tasks* $:=$ findSemanticEquivalenceTasks(*Task*)
for each *task* **in** *Tasks*
  if *task* $\notin$ *visitedTask* **then**
    $as := \emptyset$
    for each *step* **in** *task*
      $as := as \cup$ generateAS(*step*, A)
    if $as \neq \emptyset$ **and** $as \notin A$ **then return** as
      else *visitedTask* $:=$ *visitedTask* $\cup \{$*task*$\}$
return $\emptyset$

Algorithm 3 checkExcutable($\langle a_1, \ldots, a_n \rangle$)

S $:= \emptyset$
for $i := 1$ **to** n
  bound $pre(a_i)$ and $eff(a_i)$ according to $sem(as)$ and S
  if S is consistent with $pre(a_i)$
    then update S with $eff(a_i)$
    else return *False*
return *True*

Open Knowledge	AM_1	AM_2	AM_3	AM_4	AM_5	**Ratio on** AM_5
Test 1 (11615 user tasks)						
Null	1	15	17	34	50	0.43%
Tasks/Steps (11,615 rules)	13	49	89	240	307	2.64%
Test 2 (467 desires)						
Null	0	18	18	21	47	10.0%
Tasks/Steps + Help (15,020 rules)	36	78	90	103	136	29.1%

Table 4: Experimental results

$AM_4 = \{$*move, find, pick up, put down*$\}$, $AM_5 = \{$*move, find, pick up, put down, open, close*$\}$, were chosen in order to examine the impact of different action capabilities of a robot on overall performance. Neither open knowledge nor background knowledge for semantic equivalence was used in the first round of either experiment.

The experimental results are shown in Table 4. On every action model in each round, the number of tasks or desires that were fulfilled is listed in the table. In addition, the percentages of fulfilled tasks or desires with respect to the size of the test sets on AM_5 are listed in the last column. The performance on the first test set increased a little because the open knowledge used was very sparse. In fact, each rule $t \leftarrow s$ from the *Tasks/Steps* table is taken as the definition of the task t. The only open knowledge used in this test was the definitions of these tasks. Therefore, there were the same number of tasks and rules in Test 1.

The performance on the second test set increased from 10% to 29.1%, suggesting that the capability of a robot could be remarkably improved by making use of a moderate amount of open knowledge. The open knowledge used in this test was much less sparse than that in Test 1. Besides rules from the *Tasks/Steps* table, 3405 rules from the *Help* table were also used in Test 2, with each mapping a desire to a task. In addition, knowledge of synonyms from WordNet was used in the second round of both tests.

The experiments were run on a 4-core 2.8GHz Core i5 with 4G of RAM. The first round of Test 1 was completed in 9.94 seconds and the second round 3,654.281 seconds. Either round of Test 2 was completed in 1 second. The computational efficiency was acceptable for a single task or desire according to the real-time requirement of robot planning.

6 Conclusion

We are used to a notion of Knowledge Base under the implicit "closeness" assumption that a knowledge base will ideally provide sufficient knowledge for the expected domain, though it is extremely hard to develop such an ideal knowledge base for a large domain in real-world applications. The increasing development and accessibility of open-source knowledge resources bring us new opportunities to break through the difficulty. When an autonomous robot/agent employs open-source knowledge resources instead of a classical knowledge base, the closeness assumption must be abandoned. Consequently, the autonomous identification, acquisition, and utilization of open knowledge become challenging research issues. Our observations from this effort to meet the challenges are summarized as follows. First, a formalization of the subject can be established around the key concept of knowledge gaps and there is a tractable case for computing knowledge rehabilitation. Second, based on the insights from the formalized investigation, efficient techniques for translating and planning with semi-structured open-knowledge rules can be developed. Third, the underspecifiedness of open knowledge rules can be resolved to some extent. Finally, although this work is a first step towards the goal, the encouraging experimental results show that further progress would be expected in the future.

Acknowledgements

This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150 and the Natural Science Foundation of China under grant 60745002 and 61175057, as well as the USTC Key Direction Project and the USTC 985 Project. The authors are grateful to Michael Gelfond, Daniele Nardi, Victor Lesser, Peter Stone, Manuela Veloso, Mary-Anne Williams and Shlomo Zilberstein for discussions about research issues in this work.

References

[Burgard *et al.*, 1999] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.

[Cantrell *et al.*, 2012] R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz. Tell me when and why to do it!: run-time planner model updates via natural language instruction. In *Proceedings of the 7th annual ACM/IEEE international conference on Human-Robot Interaction (HRI-12)*, pages 471–478, 2012.

[Chen *et al.*, 2010] X. Chen, J. Ji, J. Jiang, G. Jin, and F. Wang. Developing high-level cognitive functions for service robots. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-10)*, pages 989–996, 2010.

[Chen *et al.*, 2012] Xiaoping Chen, Jiongkun Xie, Jianmin Ji, and Zhiqiang Sui. Toward open knowledge enabling for human-robot interaction. *Journal of Human-Robot Interaction*, 1(2):100–117, 2012.

[Collins, 2002] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, pages 1–8, 2002.

[Console *et al.*, 1991] Luca Console, Daniele Theseider Dupré, and Pietro Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[de Marneffe and Manning, 2008] Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *Proceedings of the COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*, pages 1–8, 2008.

[de Marneffe *et al.*, 2006] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-06)*, pages 449–454, 2006.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Nicola Leone. Semantics and complexity of abduction from default theories. *Artificial Intelligence*, 90(1):177–223, 1997.

[Erol *et al.*, 1995] Kutluhan Erol, James Hendler, and Dana S Nau. Htn planning: Complexity and expressivity. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-95)*, pages 1123–1123, 1995.

[Fong *et al.*, 2003] T. Fong, C. Thorpe, and C. Baur. Robot, asker of questions. *Robotics and Autonomous systems*, 42(3):235–243, 2003.

[Giunchiglia *et al.*, 2004] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153(1-2):49–104, 2004.

[Goldwasser and Roth, 2011] Dan Goldwasser and Dan Roth. Learning from natural instructions. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 1794–1800, 2011.

[Gupta and Kochenderfer, 2004] R. Gupta and M.J. Kochenderfer. Common sense data acquisition for indoor mobile robots. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, pages 605–610, 2004.

[Keller *et al.*, 2010] Thomas Keller, Patrick Eyerich, and Bernhard Nebel. Task planning for an autonomous service robot. In *Proceedings of the 33rd Annual German Conference on AI (KI-10)*, pages 358–365, 2010.

[Kruijff *et al.*, 2010] Geert-Jan M Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, Ivana Kruijff-Korbayová, and Nick Hawes. Situated dialogue processing for human-robot interaction. In *Cognitive Systems*, pages 311–364. Springer, 2010.

[Lemaignan *et al.*, 2012a] Séverin Lemaignan, Raquel Ros, E Akin Sisbot, Rachid Alami, and Michael Beetz. Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *International Journal of Social Robotics*, 4(2):181–199, 2012.

[Lemaignan *et al.*, 2012b] Séverin Lemaignan, Raquel Ros, Emrah Akin Sisbot, Rachid Alami, and Michael Beetz. Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *I. J. Social Robotics*, 4(2):181–199, 2012.

[McCain and Turner, 1997] N. McCain and H. Turner. Causal theories of action and change. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*, pages 460–465, 1997.

[Mohan *et al.*, 2012] Shiwali Mohan, Aaron Mininger, James Kirk, and John E Laird. Learning grounded language through situated interactive instruction. In *Papers from Robots Learning Interactively from Human Teachers (AAAI Fall Symposium Series)*, pages 30–37, 2012.

[Reiter, 2001] R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. Cambridge Univ Press, 2001.

[Rosenthal *et al.*, 2010] Stephanie Rosenthal, Joydeep Biswas, and Manuela Veloso. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-10)*, pages 915–922, 2010.

[Schiffer *et al.*, 2012] Stefan Schiffer, Alexander Ferrein, and Gerhard Lakemeyer. Caesar: an intelligent domestic service robot. *Intelligent Service Robotics*, 5(4):259–273, 2012.

[Talamadupula *et al.*, 2010] Kartik Talamadupula, J. Benton, Subbarao Kambhampati, Paul W. Schermerhorn, and Matthias Scheutz. Planning for human-robot teaming in open worlds. *ACM TIST*, 1(2):1–14, 2010.

[Van Harmelen *et al.*, 2008] Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of knowledge representation*. Elsevier Science, 2008.

[Xie *et al.*, 2012] Jiongkun Xie, Xiaoping Chen, JianminJi, and Sui Zhiqiang. Multi-mode natural language processing for extracting open knowledge. In *Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Intelligent Agent Technology (IAT-12)*, 2012.