Filling Knowledge Gaps in Human-Robot Interaction Using Rewritten Knowledge of Common Verbs

(Extended Abstract)

Dongcai Lu*, Jianmin Ji*, Xiaoping Chen*, and Jiangchuan Liu
University of Science and Technology of China, Hefei, P.R. China
ludc@mail.ustc.edu.cn, {jianmin,xpchen}@ustc.edu.cn, jdk@mail.ustc.edu.cn

ABSTRACT

In this paper, we present an approach to representing a core part of the knowledge consists of semantic information of common verbs from semantic dictionaries. We provide a meta-language as the representation framework for the rewritten knowledge of common verbs and their corresponding user tasks. The meta-language is interpreted based on transition systems, which can be realized on various formalizations such as situation calculus, action languages, and answer set planning. We realize the approach based on answer set planning. Moreover, we provide empirical evidence showing that HRI may significantly benefit from the rewritten knowledge and remarkable performance improvement compared to previous work.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence

Keywords

Human-Robot Interaction, Common Verbs, Knowledge Representation, Task Planning

1. INTRODUCTION

Extensive knowledge about naturally expressed tasks is needed for filling knowledge gaps in HRI, which normally provides descriptions of tasks or instructions on how to accomplish tasks. As observed in previous efforts toward enabling OMICS database [3] for robot task planning [1], common verbs become a bottleneck of utilizing existing open knowledge for task planning. This can be regarded as a knowledge gap in HRI. To attack the bottleneck, definitions of common verbs should be extracted from dictionaries or similar sources and these definitions should be rewritten into some representation processable by robots, which can fill the knowledge gap in common verbs. We provide a meta-language as the representation framework for the rewritten knowledge of common verbs and corresponding user tasks. The meta-language is interpreted based on transition systems, which implemented on answer set planning [4].

*The first three authors are corresponding authors

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

2. META-LANGUAGE

We take *FrameNet* [1], a digital dictionary providing rich semantic information of verbs, as an example to extract and represent common verbs' definitions in the meta-language, provide a translation from expressions in the meta-language to ASP rules, by which the meta-language is realized based on answer set planning. At last, we are developing a formalized version of *FrameNet*, called *Re-FrameNet* [2]. In *Re-FrameNet*, a Frame in *FrameNet* is formalized as a 'meta-task', which is re-defined by a set of precondition, postcondition, invariant, and/or steps over semantic roles of the meta-task. The definition of a meta-task specifies the common semantic structure of all action verbs in the corresponding Frame. For example, we express these knowledge in meta-language and define a meta-task *task-bringing* like this:

(**define** (**meta-task** put-placing
  (**:parameters** *?Agent ?Theme ?Source ?Goal*))
  (**:precondition** ...)
  (**:postcondition** ...)
  (**:invariant** ...))

3. PLANNING WITH THE KNOWLEDGE

We employ a three-phase procedure to translate a natural language instruction or piece of knowledge expressed in natural language into the internal representation that can be handled by our planner. First, a Stanford parser [3] is used to retrieve the syntactic structure of the instruction. Second, a meta-task is identified as the "semantic template" of the instruction, according to the action verb of the sentence. In this paper, we assume that every instruction represents just one meta-task, and we draw support from a Frame-semantic parser SEMAFOR [2] in this phase. After the meta-task is identified, its semantic roles must be filled in with the corresponding entities (expressed by nouns) in the sentence. We fill the semantic roles in the instruction using heuristic rules. At last, the single instruction *take food out of refrigerator* is interpreted as an instantiated meta-task of *take-removing* as follows

(**define** (**meta-task** take-removing
  (**:parameters** `robot food refrigerator`))
  ...)

[1]https://framenet.icsi.berkeley.edu/fndrupal/
[2]http://ai.ustc.edu.cn/en/research/reframenet.php
[3]http://nlp.stanford.edu/software/lex-parser.shtml

3.1 Algorithms

We developed a set of algorithms to plan with the rewritten knowledge over two test sets consisting of 11885 user tasks and 467 user desires collected from OMICS. Algorithm 1 is the main algorithm for planning with the rewritten knowledge over the 11885 user tasks test. Its weakened versions were used for planning without the knowledge. For example, operations of 'semantic equivalence' were not used and substituted by that of 'syntactic equal' in planning without the rewritten knowledge.

Algorithm 1 overallPlan(task t)

```
/* generate a plan p for task t from Tasks/Steps */
initiate worldmodel and p
if t is visited then return (False) endif
for each step of t do
  s := parseFrame( step, worldmodel, p)
  update worldmodel according to s
  Res := clasPlan( s )
  if Res ≠ null then
    save Res to p and update worldmodel by Res
    continue
  end if
  while there is a new t' from Tasks/Steps semantically
  equivalent to step do
    if not overallPlan(t') then
      regress worldmodel and p
    else
      break
    end if
  end while
  if s is not solved then return (False) endif
end for
return(True)
```

4. EXPERIMENTAL RESULTS

The experiments aimed to investigate the performance of our meta-language framework when different bodies of online knowledge were used, and analyze the main factors that affect the performance. Test 1 was conducted on 11885 user tasks from the *Tasks/Steps* table of OMICS, consisted of three rounds. In the first round of Test 1, only the definitions of these 11885 tasks from the *Tasks/Steps* table and a small action model *AM* were used. *AM* contained only 6 primitive actions: *move, find, pick_up, put_down, open*, and *close*. Synonymy knowledge from *WordNet* was added into the second to third round of Test 1 and rewritten knowledge from *Re-FrameNet* into the third round.

Table 1 shows the experimental results of Test 1. The second line shows the numbers of tasks that were successfully planned in the three rounds of Test 1. The third line shows the percentages of successfully planned tasks with respect to the total number of tested tasks, 11885. The fourth line of Table 1 shows the percentages of successfully planned tasks with respect to the number of tasks that actually entered planning. One can see that the overall performance improved **twice** when *Re-FrameNet* was used.

We also checked the correctness of the successfully planned tasks in Test 1. Of course, success does not imply correctness. Since there are no ground truth data for OMICS, we drew 80 samples randomly from 274 and 652 successful tasks

Table 1: Experimental results over 11885 user tasks.

Knowledge used (Tasks/Steps+)	**AM**	**AM+ WordNet** (baseline)	**AM+WordNet +RFN**
Numb. success	238	274	**652**
Success percent	2.00% (1)	2.30% (**+15%**)	5.48% (**+174%**)
Success percent wrt ParseFrame	4.92% (1)	5.66% (**+15%**)	13.49% (**+174%**)
Correctness percent		82.50%	63.75%
Correct plans (Improvement)		226 (1)	416 (**+84%**)

Table 2: Experimental results over 467 user desires.

Knowledge used (Tasks/Steps+)	**AM**	**AM+ WordNet** (baseline)	**AM+WordNet +RFN**
Numb. success	144	173	**364**
Success percent	30.84%	37.04%	**77.94%**
Correct success percent		30.59%	**49.69%**

in the last two rounds, respectively, and verified them manually. It turned out that 66 and 51 samples were correct. The fifth line of Table 1 shows the results.The correctness percent decreased when *Re-FrameNet* was used; but the number of correctly planned tasks still increased remarkable, as shown in the last line of the table.

Now we report Test 2 on 467 user desires from the *Help* table of OMICS. Since a *Help* tuple maps a user desire to a task, the algorithms for Test 2 were developed based on those for Test 1, by just adding a higher-layer to handle the mapping between desires and tasks. This indicates that the hierarchism of user instructions can ease the development of HRI systems significantly. From the experimental results (Table 2), one can see that the success percents were higher than every corresponding round of Test 1. This is because of the fact that a desire can be met by various tasks, although these tasks are different one another.

5. ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China under grant 61175057 and the USTC Key-Direction Research Fund under grant WK0110000028.

REFERENCES

[1] X. Chen, J. Xie, J. Ji, and Z. Sui. Toward open knowledge enabling for human-robot interaction. *Journal of Human-Robot Interaction*, 1(2):100–117, 2012.

[2] D. Das, D. Chen, N. Schneider, and N. A. Smith. Frame-Semantic Parsing. *Computational Linguistics*, 39(4):1–47, 2012.

[3] R. Gupta and M. 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.

[4] V. Lifschitz. Answer set planning. In *Proceedings of the 1999 International Conference on Logic Programming (ICLP-99)*, pages 23–37, 1999.