International Journal of Automation and Computing

15(5), October 2018, 582-592
DOI: 10.1007/s11633-018-1128-9

Learning to Transform Service Instructions into Actions with Reinforcement Learning and Knowledge Base

Meng-Yang Zhang[1,2] Guo-Hui Tian[1,2] Ci-Ci Li[1,2] Jing Gong[1]

[1] School of Control Science and Engineering, Shandong University, Jinan 253000, China
[2] Shenzhen Research Institute, Shandong University, Shenzhen 518000, China

Abstract: In order to improve the learning ability of robots, we present a reinforcement learning approach with a knowledge base for mapping natural language instructions to executable action sequences. A simulated platform with physical engine is built as interactive environment. Based on the knowledge base, a reward function with immediate rewards and delayed rewards is designed to handle sparse reward problems. Also, a list of object states is produced by retrieving the knowledge base, as a standard to define the quality of action sequences. Experimental results demonstrate that our approach yields good performance on accuracy of action sequences production.

Keywords: Natural language, robot, knowledge base, reinforcement learning, object state.

1 Introduction

In recent years, applications of robots exist in a lot of areas, such as industry, healthcare, education, social life, etc. It has been extensively believed that robots can increase the work efficiency, and bring convenience to people′s life. However, the ability of robots to perform complex tasks is limited, especially when it comes to tasks about home service.

As a result, there has been a growing interest in developing and investigating methods to improve robots′ operating ability. Previous works on this can be divided into two types. One type is to extract elements from the obtained information with manually designed rules and provide executable strategies[1, 2], referred to as rule construction. Another type, the mainstream for modifying robots′ ability, is knowledge construction[3–5], which can work as an associative mean by constructing knowledge bases specific to services, as the lack of relevant knowledge is the main cause preventing robots from completing service requirements. The knowledge bases can be constructed with logic[6], Stanford Research Institute Problem Solver (STRIPS)[7], planning definition domain language (PDDL)[8], probability[9], or other representations. Researchers of knowledge representation model scenarios as a dynamic system in a knowledge base and perform control, prediction and analysis tasks by inferring solutions purely based on this model.

Although the ability of robots for services can be modified with above approaches, there is a strict requirement on manual effort, as the construction of service rules or knowledge is a labour intensive process. And thus, attentions have been focused on increasing robots′ operating ability with available resources for decreasing human consumption.

Text is the universal information carrier stored and shared on the internet. It can be accessed easily, so exploiting text information for increasing the ability of robots is critical. Hameed[10] built a database with a record of information about personal habits so as to assist robots in understanding the intention of human beings. Tenorth et al.[11, 12] proposed a method to construct executable plans for robots by parsing and representing the online text information with logical expression and sentence processing. It has been proved that service instructions as guide can increase the ability of robots for service operation, but the process of parsing and exploiting texts is complicated and has to be done under supervision of researchers. So allowing robots to learn automatically is promising.

1.1 Related works

Inspired by recent advances in deep learning[13–16], combining deep learning with reinforcement learning has made significant progress[17–20]. Reinforcement learning aiming to maximize the rewards in the long term can enforce the learning ability of robot[21]. In natural language processing, reinforcement learning has been applied successfully.

He et al.[22] proposed a learning model based on deep

Research Article
Special Issue on Intelligent Control and Computing in Advanced Robotics
Manuscript received January 19, 2018; accepted March 20, 2018; published online May 30, 2018
Recommended by Guest Editor Jun-Zhi Yu

reinforcement learning, referred to as deep reinforcement relevance network-bidirectional long short term memory (DRRN-BiLSTM). With the model, text strings can be processed and mapped to a combinatorial, natural language action space. Sentences can be represented in a format of tree structure[23, 24]. Based on this conception, Bowman et al.[25] proposed a novel model with reinforcement learning, and the model can yield good performance on parsing and sentence understanding.

Also approaches taking natural language information as guide in reinforcement learning were applied in fields of video games and information retrieval[26, 27], and yielded good performance. Similar to our goal in this paper, Branavan et al.[28] proposed a method that maps instructions into actions, but the interactive environment and involved vocabulary is relatively simple, which is not enough for performing complex tasks.

In the above-mentioned methods, there is a unified standard for judging the final result, and the environment involved is relatively simple, but they are still unsatisfactory in dealing situations with complex environment and ambiguous criterion.

1.2 Proposed approach

In this paper, we proposed a method for transforming instructions related to home service into action sequences, with a knowledge base and reinforcement learning.

The sparse rewards problem may occur in reinforcement learning when reward functions are simple and not enough to promote parameter convergence[29–31]. And complex environment, especially family environment, can cause sparse rewards problems, which makes parameters difficult to converge and lowers the accuracy of results.

To address this problem, we built a hierarchical knowledge base of home service, and the knowledge base can provide service information on both object and service level. Based on the knowledge base, a reward function with immediate rewards and delayed rewards is designed. Immediate rewards are produced along with rules guiding the direction of producing proper action sequences, and delayed rewards are given by estimating the final result of task operation.

Unlike video games with clear results, win or failure, in virtual environment as in [32–34], there is no uniform standard to measure the effect of home service operation. To deal with this, a standard based on object states is designed for judging the operating result of home service. The information related to object states can be obtained by retrieving the knowledge base.

Inspired by the view that the feedback of the environment from direct communication can be used for online decision making immediately[35], we build a simulated environment with physical information from the knowledge base, as is shown in Fig. 1. Actions can be chosen and operated in the simulated environment, and there is states transition along with action execution.

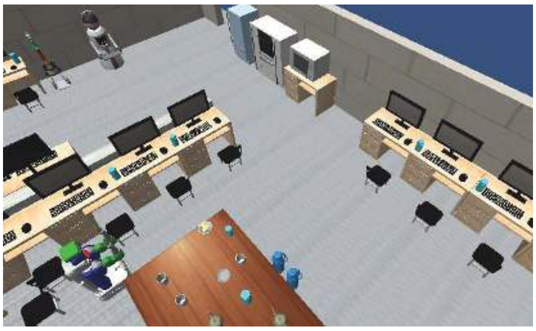


Fig. 1 Simulated environment for reinforcement learning. The number of objects related to home service is large. As the platform is used for testifying the logic of produced action sequences, we focus more attention on the intrinsic characteristics of objects than the visual reality.

There are two main contributions in this paper. One is that we present a standard for estimating home tasks execution by introducing object states. Another is the application of knowledge base, which is essential for environment construction in reinforcement learning and object states acquisition.

The remainder of the paper is organized as follows. The overview of the proposed method is introduced in Section 2. The process of semantic fragment mapping is described in Section 3. The application of the knowledge base is presented in Section 4. And the implementation of reinforcement learning is shown in Section 5. The experiments and conclusion are presented in Sections 6 and 7, respectively.

2 Framework overview

The framework is composed of information source related to home service, a knowledge base and the dynamic simulation platform, as illustrated in Fig. 2.

The website, wikiHow, is chosen as information source. It contains information of a variety of services, and its content is represented in semi-structured format, which means the information is expressed in steps, so it can facilitate the process of extracting information.

Firstly, the content from information source is stored in documents, each document corresponds to a service. With the process of semantic fragment mapping, information from documents is transformed into initial action sequences with no identification on relationships between objects and operators. During the mapping process, the essential factors in documents are mapped to corresponding action functions and parameters complying no rules but the word meaning.

Then, the initial action sequences are sent to the dynamic simulation platform for logic verification. The platform plays a role as the interactive environment in reinforcement learning. It consists of virtual scene modeled based on our laboratory, object models related to home service, and a library of action functions. As basic elements, object models have characteristics of physical

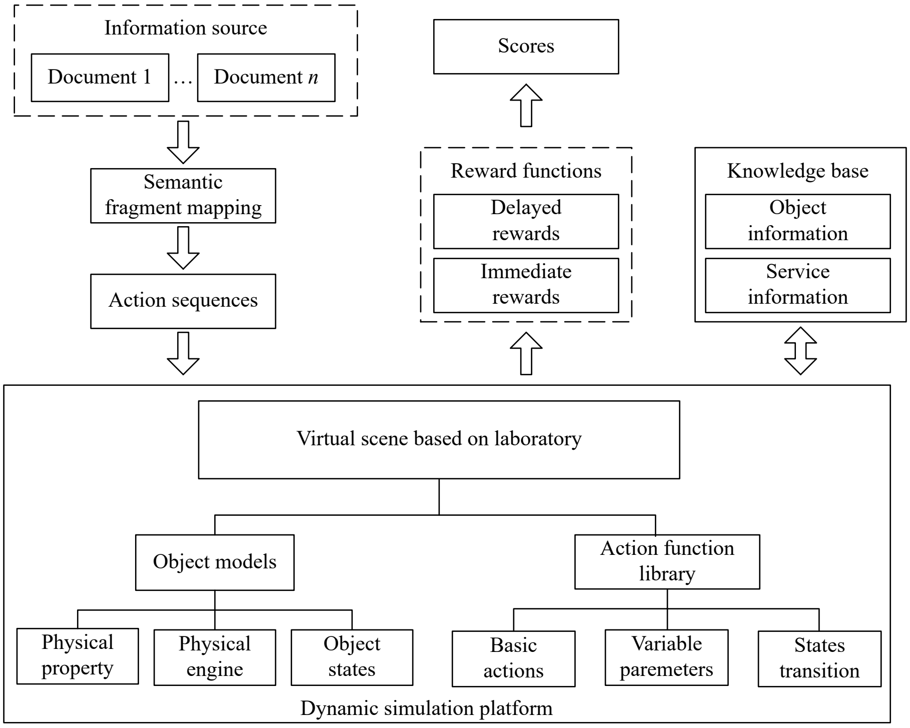


Fig. 2 Framework of the proposed method. The goal of our deep learning model is to raise the ability of robots for home service by acquiring related knowledge from natural language information about family service. The information related to home service is taken as inputs, and action sequences which can be used or referenced by robots are generated as the output. The model can deal with complex family environment and produce actions oriented to home service.

property, physical engine and object states. Physical property indicates objects′ size, weight, color, etc. Object states are current states. For example, the object state of a cup full of water is full. These two kinds of information can be obtained by retrieving a knowledge base. The physical effects of object models can be reflected with physical engine, which is useful for simulating service execution. The action function library includes basic actions such as grab, open, lay down, etc. There is states transition when an action function is operated, and we take object states composition as a standard to estimate service execution. Also we take into consideration the problem of variable parameters which is caused by the sentence structures.

Finally, a reward function with immediate rewards and delayed rewards is designed to produce scores on the effect of initial action sequences. Regarded as a task, each document is composed of subtasks, including sub-actions, relevant objects and object states as is illustrated in Fig. 5. When a subtask is completed, the immediate reward is given. And the value of delayed rewards stands for completion of the task, which is based on object states.

The knowledge base is constructed to provide information about object and service. Object information contains physical information corresponding to object models, which can be obtained for setting model parameters. Service information takes home services as tasks, each task includes objects involved in the corresponding service, and object states. Application of knowledge base is essential for construction of simulated platform and reward function design.

3 Semantic fragment mapping

The process of semantic fragment mapping is illustrated in Fig. 3. Sentences from information source are sent to the mapping layer for semantic parsing and sense disambiguation. After that, the processed information is mapped to corresponding names of functions in an action function library. Then, the selected functions are performed in the dynamic simulation platform.

The information source is divided into documents related to home service. Each document is composed of sentences and represents a service topic.

The mapping layer is a bridge between information source and the action function library, through which initial action sequences can be obtained by mapping sentences to action functions. It consists of Stanford Lexical Parser[36], Counter and WordNet. Stanford Lexical Parser is applied here for semantic parsing by extracting noun and verb-phrases from sentences, and the parsed fragments are stored in a list as a candidate set. WordNet is a lexical database which can provide synonym collection. With it, words with similar meaning can be mapped to a unified vocabulary, so as to reduce complexity of information processing. And Counter is used to record the number of nouns in a sentence, indicating the mapping de-

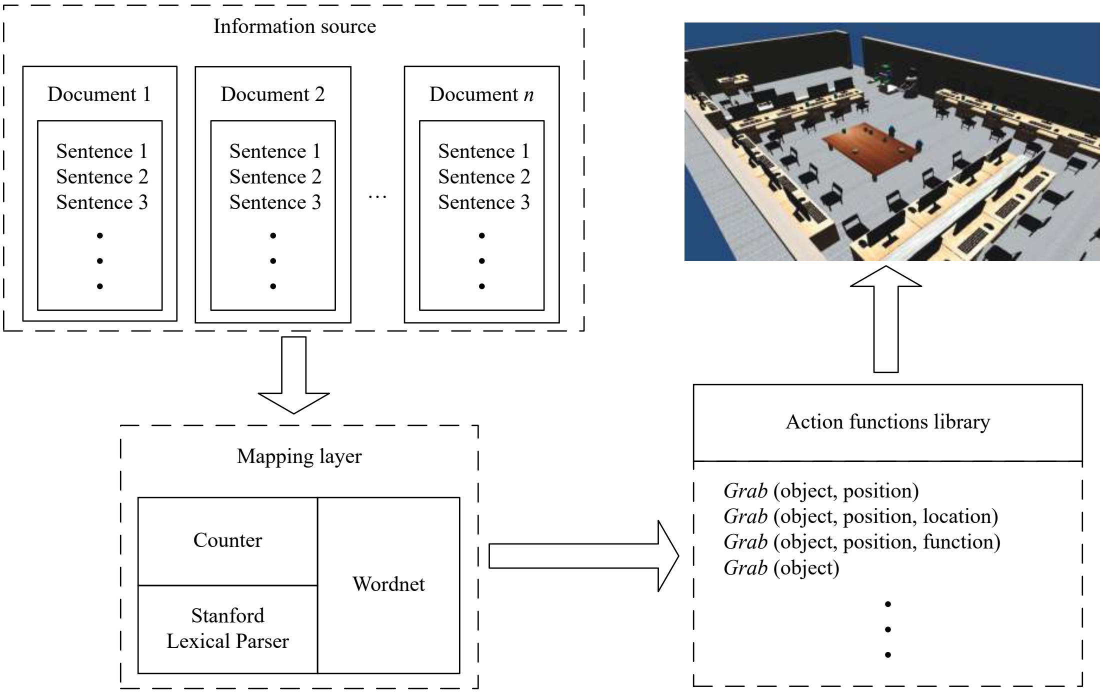


Fig. 3 Process for semantic fragment mapping. The key elements of a sentence are extracted and mapped to functions in an action library. Parameters corresponding to objects are chosen randomly without identification of object relationships.

gree of sentences as a factor for function selection.

During semantic fragment mapping, we take into consideration two problems, the scale of vocabulary in information source, and action selection specific to the same behaviour.

Although the area of research is limited to home service, the scale of vocabulary indicating actions and objects is still large, which makes action functions construction difficult. Thus, we construct lists of synonyms in order to reduce the vocabulary scale. For example, the words, take and get, have the same meaning for grabbing, so both of the words will be mapped to the same action function: *Grab*.

Another problem is action selection with variable parameters. The structure of sentences is complicated, and the meaning of sentences can be different when the number of nouns in a sentence changes. As illustrated in Fig. 3, there are four kinds of action functions corresponding to the word, grab, which are *Grab*(*obj*, *pos*), *Grab*(*obj*, *pos*, *loc*), *Grab*(*obj*, *pos*, *func*) and *Grab*(*obj*). The number of parameters in action functions reflects different structures of sentences. For example, *Grab*(*obj*) means the robot should grab an object labeled as *obj*, and the position of the object should be obtainded by the robot itself. And the function, *Grab*(*obj*, *pos*), is different from *Grab*(*obj*), as the information indicating object position can be acquired from sentences. Also *Grab*(*obj*, *pos*, *loc*) means the robot grabs the object, *obj*, at a specific position, *pos*, and takes the object to a place, *loc*. Therefore, the Counter is used to record the number of nouns in a sentence which is taken as a factor of choosing proper action functions.

Through the mapping layer, sentences are transformed into initial action sequences. As the executing orders of actions and parameters are not considered, the sequences need to be sent to the dynamic simulation platform to test the reasonability.

4 Application of knowledge base

A knowledge base is built to provide information on both object and service level. Based on the knowledge base, reinforcement learning is implemented to transform service instructions into action sequences.

As illustrated in Fig. 4, the knowledge base is constructed in hierarchy and divided into object level and service level.

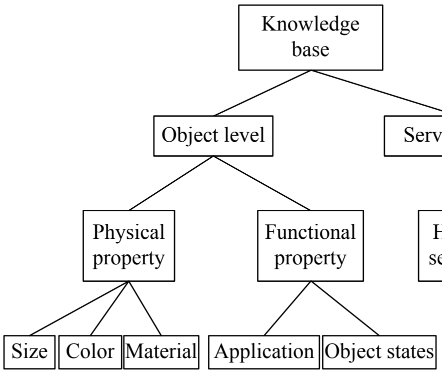


Fig. 4 Structure of the knowledge base about home service. It contains information on both the object level and the service level. Information on object level includes intrinsic characteristics of objects, and service tasks are stored in service level.

The knowledge on object level includes physical property and functional property. The physical property denotes information of objects about their size, colour and material, which are used for model construction by setting parameters of models based on the information. And the functional property involves information about the application and state of objects. The application information is not a description stating the usage of the object, but a network that links the relevant nouns and verbs with the object, which can be used to design rules of the reward functions. The object states indicate the current state of objects on functional aspect. For example, when an empty cup is filled with water, its state has changed from empty to full. Thus, the state of a cup is represented below:

$$[cup : (containable, empty)].$$

The knowledge on service level takes home service tasks as units, as illustrated in Fig. 5. Each task can be separated into subtasks representing sub-actions, and the relevant objects and their states are at the bottom of the knowledge. For instance, cleaning room is a task, with subtasks such as wiping the table, sweeping the floor, mopping the floor, etc. Each subtask involves task-relevant objects. Objects in the task of wiping a table include a table and a rag, while a broom and a dustbin are needed for sweeping the floor. The object states stored on service level are the final states after task execution, so a list of final states according to specific tasks can be produced by retrieving the knowledge base on service level.

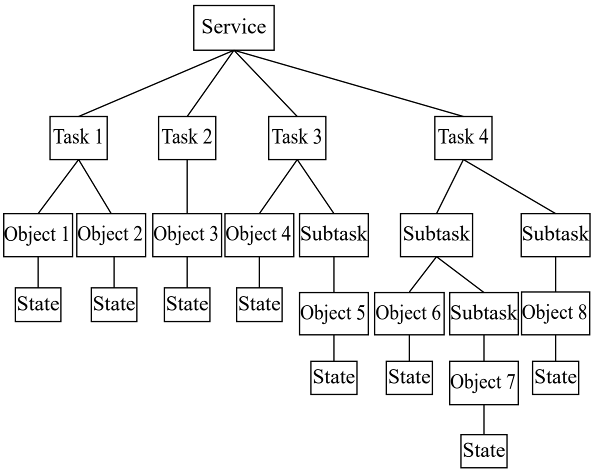


Fig. 5 Service level of knowledge base. It takes tasks or subtasks as units. Elements that make up each unit are object states.

5 Reinforcement learning

Home environment is complex. There is not a unified standard to estimate the task execution of home service, as action sequences specific to tasks are variable. In order to implement reinforcement learning on home service, it is crucial to establish a unified standard.

5.1 Acquisition of object states

We present a standard taking object states as factors for judging the final result of service execution, so object states acquisition is essential. The acquisition process is illustrated in Fig. 6.

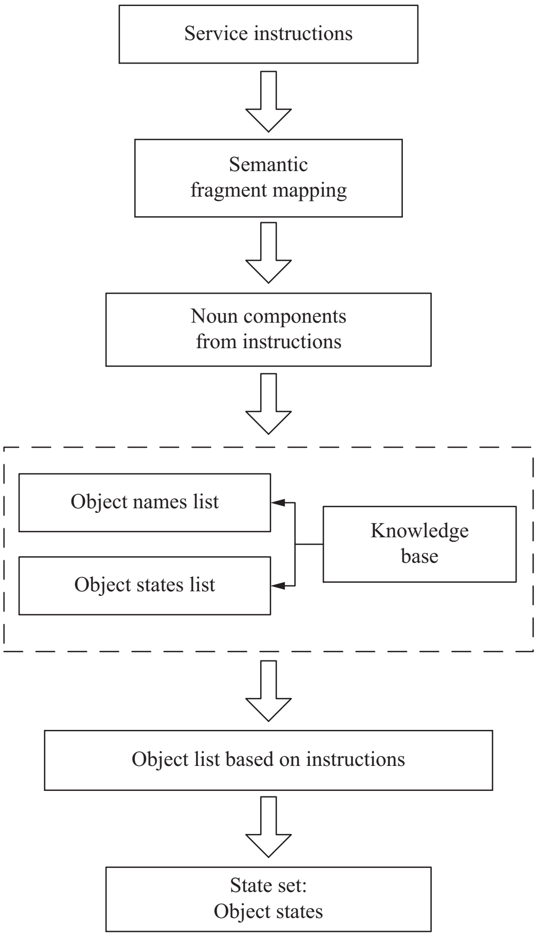


Fig. 6 Process of obtaining object states. Objects involved in service instructions can be obtained and attached to corresponding object states through the process. The knowledge base is a key component for the realization of the process.

Firstly, as content from information source takes the form of semi-structured expression, texts related to home service are represented in the form of a tuple: (*request*, *service*). The request part indicates the theme of service, which can be used to search for corresponding information on service level of the knowledge base. And the service part responds to service execution and is described in steps, which is the foundation of subtask construction. After semantic fragment mapping, noun components from request part are taken to match relevant tasks on service level of the knowledge base, so a list containing task-relevant objects is obtained:

$$N = [n_1, n_2, \cdots, n_m] \tag{1}$$

where N is a list of strings about object names related to tasks, and n_i indicates the basic element.

Then, object names in service part can be extracted by traversing the list N, and object names satisfying a mapping relationship are stored in another list H:

$$H = [obj_1, obj_2, \cdots, obj_n] \tag{2}$$

where obj_i is the name of object in service part.

The next step is to add object states to objects in list H by referring to the knowledge base, and the processed list is represented as H.

$$H = [obj_1 : state_1, obj_2 : state_2, \cdots, obj_h : state_h] \tag{3}$$

where $state_i$ is a tuple in a format of $(name, value)$, where $name$ indicates the name of object states and $value$ is the corresponding value of $name$. For example, the state of a cup, $state_i$, can be expressed as (*containable*: full) or (*containable*: empty).

5.2 Design of reward functions

To address the problem of sparse rewards in home environment, a two-level reward function is designed for producing logical action sequences from natural language, by setting rules to estimate the completion of subtasks and the final task.

The first level of the reward function, referred to as immediate rewards, is to produce scores for estimating the completion of subtasks. Immediate rewards can be divided into two parts. One part inspired by the proposal in [12, 13] is produced by mapping semantic fragments in sentences to relevant elements in simulation platform including object models and action functions. Taking a sentence as a unit, a positive score is given when nouns in the sentence can be mapped to object models, or there is a match between verbs and action functions. Also if the information represented from the mapped models and functions can be linked in the application information, there is a positive score. Another part is to estimate completion of subtasks involved in a sentence. The object states in sentence are extracted and compared with final states of objects in the knowledge base. If the result is consistent, the score is positive, or negative on the contrary.

The second level is referred to as the delayed rewards, which are used to estimate the effect of the final result.

Based on information on service level of the knowledge base, object states can be obtained so as to compare similarity with the object states after executing action sequences.

Taking documents as units, we first get object states, S_O, from a document by traversing the knowledge base.

$$S_O = [state_1, state_2, \cdots, state_w] \tag{4}$$

where $state_i$ is the corresponding state of objects, and w is the total number of involved objects.

Then S_P, object states after operating actions, is constructed with the same order and number of objects as S_O. Test the object states with the equation below:

$$F(S_O, S_P) > K \tag{5}$$

where F is a function outputting the proportion of S_P in S_O, K is a threshold in interval (0, 1). When the output of function F is larger than K, the produced action sequences are reasonable.

5.3 Implementation of reinforcement learning

We adopt policy gradient algorithm as a way of getting the optimal policy $p = (a|s, \theta)$ which is used to get optimal action composition by tuning parameters θ for maximum expected rewards[37], where a represents the chosen actions and s is the state.

Firstly, the data set is represented as follows.

$$D = [d_1, d_2, \cdots, d_N] \tag{6}$$

$$d_i = [u_1, u_2, \cdots, u_M] \tag{7}$$

where d is a document representing specific service tasks, and u_i is a sentence in a document.

The information in d_i is extracted and mapped into action sequences $a = [a_1, a_2, \cdots]$, the action a_i is representted in a form of triple, $a_i = (fn, par, W')$, where fn represents the name of action functions in simulation platform, par are parameters in fn, and W' specifies the mapping words from documents.

Then, the state $s = (\varepsilon, d, j, W)$ from documents to action sequences is constructed, where ε is the composition of current object states in simulation environment, d is the document containing the corresponding service information, j indicates the index of sentences in a document and W contains the words mapped from the produced action sequences. Following the distribution $p = (s'|s, a)$, a new state s' will be produced when an action a is executed at the state s.

Based on those information, the value function is built as

$$V_\theta(s) = E_{p(h|\theta)}[r(h)]. \tag{8}$$

The history $h = (s_0, a_0, \cdots, s_n)$ makes a record on the executed actions and experienced states during processing the document, and $r(h)$ is the reward of the history h.

Finally, the policy gradient algorithm is employed to maximize the parameter θ with following rules:

$$\frac{\partial}{\partial\theta}V_{\theta}(s)=E_{p(h|\theta)}[r(h)\sum_{t}\frac{\partial}{\partial\theta}\log p(a_t|s_t;\theta)] \qquad (9)$$

where $p=(a|s,\theta)$ is a softmax function, and its derivative of logarithmic form is represented like this

$$\frac{\partial}{\partial\theta}\log p(a|s;\theta)=\varPhi(s,a)-\sum_{a'}\varPhi(s,a')p(a'|s;\theta) \qquad (10)$$

where $\varPhi(s,a)$ is the feature representation and we get samples by using the distribution $p(h|\theta)$ for obtaining the gradient of the value function. The parameter θ is updated with the following rules:

$$\varDelta=\sum_{t}(\varPhi(s_t,a_t)-\sum_{a'}\varPhi(s_t,a')p(a'|s_t;\theta)) \qquad (11)$$

$$\theta=\theta+r(h)\varDelta. \qquad (12)$$

6 Experiment and discussion

Our main objective for producing action sequences is to find a way to identify the relationship of semantic elements in a sentence effectively. In this section, we first collect information and build necessary dataset including the home service information and the knowledge base. Then, the interactive environment, the simulated platform, is constructed based on the dataset. We evaluate our method with baseline methods from both aspects of the convergence rate and correctness. Finally, other problems about the production of action sequences will be discussed.

6.1 Dataset

We choose wikiHow, a website which provides information related to how to serve, as information source. Based on its sitemap, an XML file which makes a record on categories and path of related information, we obtain documents corresponding to home service. The specific information on documents is shown in Tables 1 and 2.

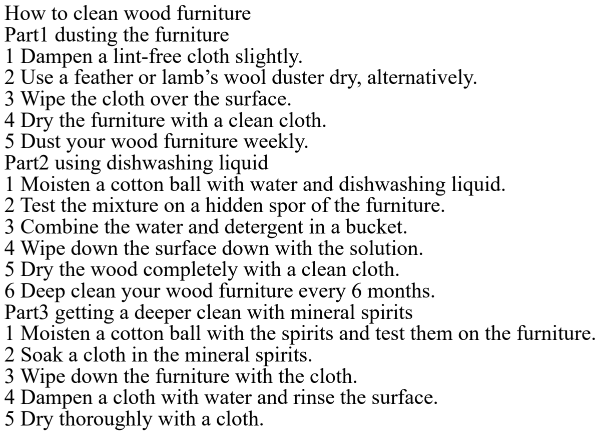
Our dataset consists of 1000 documents related to home service. Based on the sitemap, the documents can be categorized into housekeeping, house decoration, cooking and cleaning. Housekeeping contains a variety of service related to house keeping, such as how to tidy up a wardrobe, or how to clean the bathroom. House decoration gives instructions on how to make home environment comfortable, mainly focusing on item placement. Cooking and cleaning indicate execution of simple tasks, like how to cook coffee or clean a bidet. In order to reduce the complexity of information processing, the first sentence in each steps, which encapsulates the whole paragraph, is extracted and stored, while removing other information including tips and notes. The representation of information is shown in Fig. 7.

Table 1 Basic statistics of documents

Category	Number
Housekeeping	463
House decoration	123
Cooking	76
Cleaning	340

Table 2 Statistics about the dataset of home service

Total number of documents	1000
Total number of words	197361
Vocabulary size	5973
Sentences per document	21.78
Words per sentence	9.27



How to clean wood furniture
Part1 dusting the furniture
1 Dampen a lint-free cloth slightly.
2 Use a feather or lamb's wool duster dry, alternatively.
3 Wipe the cloth over the surface.
4 Dry the furniture with a clean cloth.
5 Dust your wood furniture weekly.
Part2 using dishwashing liquid
1 Moisten a cotton ball with water and dishwashing liquid.
2 Test the mixture on a hidden spor of the furniture.
3 Combine the water and detergent in a bucket.
4 Wipe down the surface down with the solution.
5 Dry the wood completely with a clean cloth.
6 Deep clean your wood furniture every 6 months.
Part3 getting a deeper clean with mineral spirits
1 Moisten a cotton ball with the spirits and test them on the furniture.
2 Soak a cloth in the mineral spirits.
3 Wipe down the furniture with the cloth.
4 Dampen a cloth with water and rinse the surface.
5 Dry thoroughly with a cloth.

Fig. 7 Information on how to clean wood furniture after removing unnecessary information. The description for task execution is separated into several parts and each part is presented in steps. The format is suitable for information processing.

The data exhibits certain qualities that make for a challenging learning problem. We extract nouns and verbs from sentences as key elements, also the vocabulary is reduced with the synsets in WordNet.

6.2 Construction of simulated platform

In order to realize real-time interaction, the simulated platform with physical engine is built with Unity 3D. The platform consists of a library of action functions and a simulated scenario with object models, as is illustrated in Fig. 8.

Action functions indicate basic behaviors from robots, which have an influence on the states of objects when executed. During the mapping process of semantic fragments, the names of action functions are assigned to verb phrases from sentences. Also, action functions with differ-

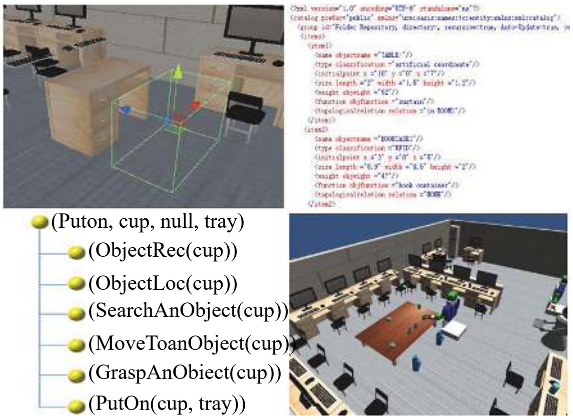


Fig. 8 Construction of simulated platform. The knowledge base can provide physical information for model construction. The library of action functions is the bridge between the agent and the environment, and action execution can lead to transition of object states.

ent parameters are considered due to the diversity of sentences representation.

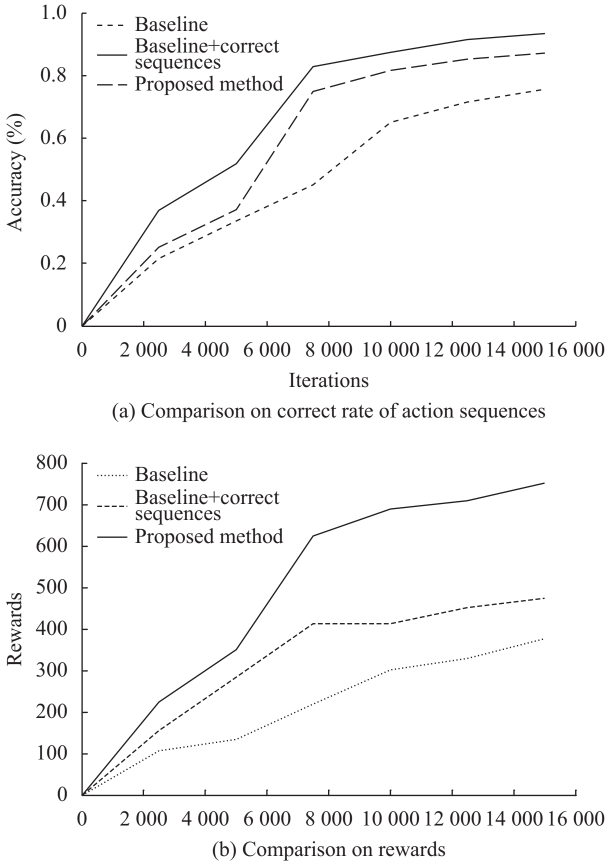
Unlike the platform constructed in [35], we decrease the visual effect reflected by the models because the objective of this platform is to test the logic of action sequences derived from instructions. Therefore, attention has been given to the size, shape and structure of the models, not the texture, in order to save computation consumption. Model construction is based on physical property information which applies to object information including colour, size, weight, etc.

In this paper, Protege 3.4.4 is used to construct the knowledge base, and we store the information of both physical and functional property with RDF (resource description framework).

6.3 Experiments

In this section, we describe experimental results along with qualitative analysis. Since the goal of the proposed method is to produce logical action sequences, metrics such as bilingual evaluation understudy (BLEU) and perplexity which are used for dialogue quality evaluation are not suitable. So we evaluate the logic of produced results with human judgments.

Comparative experiments are designed to test the validity of the proposed method. We take the method with only the policy gradient algorithm as the baseline, and the difference between the proposed method and the baseline is that the former one is integrated with immediate rewards for the sparse problem. Also, we designed a method for which the parameters are tuned by training with 500 correct action sequences constructed manually. These methods are compared in aspects of the correct rate and rewards, in order to prove the feasibility of the proposed method on home service. The experimental results are shown in Fig. 9.



(a) Comparison on correct rate of action sequences

(b) Comparison on rewards

Fig. 9 Comparison of results based on three methods. The advantages of the proposed method are stated in both aspects of correct rate and rewards.

The results illustrated in Fig. 9 (a) demonstrate that the baseline gets the lowest score on accuracy, while the method trained with the manually constructed action sequences acquires the highest accuracy. The accuracy of the proposed method is lower but close to the best one, what should be noted is that the proposed method can save human resource to get a result of good quality.

In Fig. 9 (b), we can see the baseline obtains the lowest score of rewards, and the score of the method trained with correct sequences is a little better, because these two methods take the final result as the only standard for judgment. The reward of our proposed method is the highest because of the immediate rewards for subtasks, also it indicates the tendency for convergence towards high rate of accuracy, which makes clear the guidance of immediate rewards for proper action sequences.

We also make comparison with results on different categories, as is illustrated in Fig. 10. The method trained with manual samples yields better performance, but experimental results indicate our proposed method has the same performance as the best one in areas of housekeeping, house decoration and cleaning. The correct rate of the three approaches are relatively low in cooking compared with other areas. One reason is the small number of documents in cooking area. And we ascribe another to

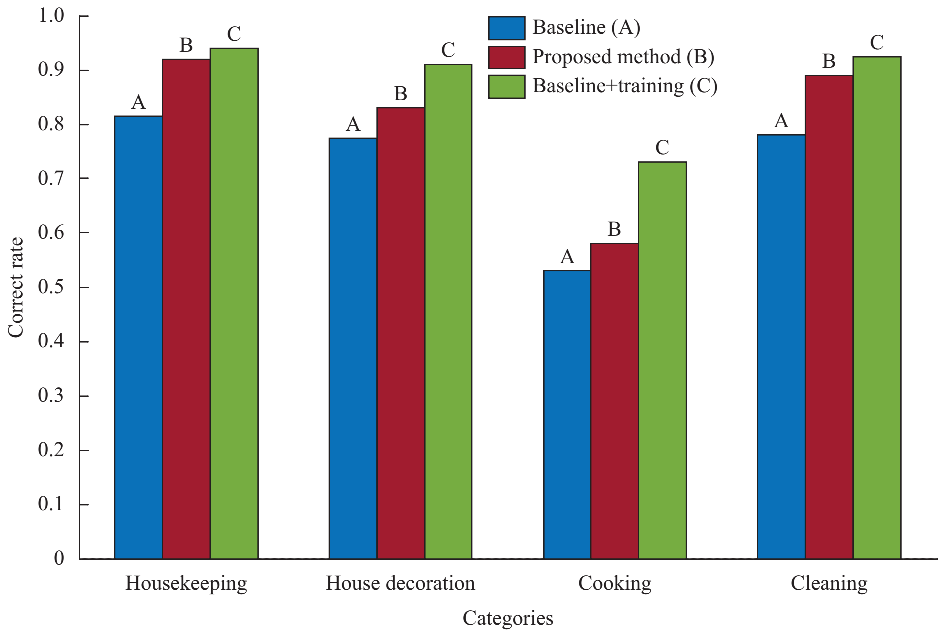


Fig. 10 Comparison of results based on three methods. The advantages of the proposed method are stated in both aspects of correct rate and rewards.

the absence of knowledge on cooking, by analyzing service categories in the knowledge base, we find the knowledge base cannot provide enough information of objects and the corresponding states. Also information on cooking includes various objects which are not as common as tools like table, chair, broom.

6.4 Discussions

With the help of the knowledge base and the simulated platform, the proposed method can exploit the self-learning ability of reinforcement learning to obtain optimal policies. Compared with traditional methods on increasing the intelligence of robots, the proposed method can save manual effort.

One aspect to be noted is the feature representation. As the paper aims to find an effective way to produce action sequences from instructions, we take the list of object states and the chosen action functions as the states for reinforcement learning, without considering image features as part of the states which are applied in [18, 26, 32, 38].

7 Conclusions

In this paper, we presented a reinforcement learning approach for inducing a mapping between instructions of home service and action sequences. Our method provides contributions in two aspects. First, we propose a way to judge the result of home service operation by taking object states as evaluation factors. Second, the knowledge base is employed as associative means for reinforcement learning implementation. The experimental results have demonstrated that the proposed method can yield good performance on producing proper action sequences.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61773239) and Shenzhen Future Industry Special Fund (No. JCYJ20160331174814755).

References

[1] W. Wang, Q. F. Zhao, T. H. Zhu. Research of natural language understanding in human-service robot interaction. *Microcomputer Applications*, vol. 3, no. 1, pp. 45–49, 2015.

[2] L. F. Shang, Z. D. Lu, H. Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, IEEE, Beijing, China, pp. 1577–1586, 2015. DOI: 10.3115/v1/P15-1152.

[3] J. M. Ji, X. P. Chen. A weighted causal theory for acquiring and utilizing open knowledge. *International Journal of Approximate Reasoning*, vol. 55, no. 9, pp. 2071–2082, 2014. DOI: 10.1016/j.ijar.2014.03.002.

[4] M. Tenorth, M. Beetz. Know rob-knowledge processing for autonomous personal robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, St. Louis, USA, pp. 4261–4266, 2009. DOI: 10.1109/IROS.2009.5354602.

[5] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. M. M. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, R. van de Molengraft. Roboearth. *IEEE Robotics and Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011. DOI: 10.1109/MRA.2011.941632.

[6] R. Reiter. *Knowledge in Action: Logical Foundations for*

Specifying and Implementing Dynamical Systems, Cambridge, USA: MIT Press, 2001.

[7] D. McDermott. The formal semantics of processes in PDDL. In *Proceedings of the 23th International Conference on Automated Planning Scheduling*, Rome, Italy, 2003.

[8] M. Fox, D. Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003. DOI: 10.1613/jair.1129.

[9] L. P. Kaelbling, M. L. Littman, A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, 1998. DOI: 10.1016/S0004-3702(98)00023-X.

[10] I. A. Hameed. Using natural language processing (NLP) for designing socially intelligent robots. In *Proceedings of Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*, IEEE, Cergy-Pontoises, France, pp. 268–269, 2016. DOI: 10.1109/DEVLRN.2016.7846830.

[11] M. Tenorth, D. Nyga, M. Beetz. Understanding and executing instructions for everyday manipulation tasks from the World Wide Web. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Anchorage, USA, pp. 1486–1491, 2010. DOI: 10.1109/ROBOT.2010.5509955.

[12] M. Tenorth, U. Klank, D. Pangercic, M. Beetz. Web-enabled robots. *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 58–68, 2011. DOI: 10.1109/MRA.2011.940993.

[13] Y. LeCun, Y. G. Bengio, G. Hinton. Deep learning. *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539.

[14] L. Deng, D. Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014. DOI: 10.1561/2000000039.

[15] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012. DOI: 10.1109/MSP.2012.2205597.

[16] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, Lake Tahoe, USA, pp. 1097–1105, 2012.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. DOI: 10.1038/nature14236.

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. DOI: 10.1038/nature16961.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra. Continuous control with deep reinforcement learning. *Computer Science*, vol. 529, no. 7587, pp. 484–489, 2015.

[20] Y. Duan, X. Chen, R. Houthooft, J. Schulman, P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on Machine Learning*, ACM, New York, USA, pp. 1329–1338, 2016.

[21] R. S. Sutton, A. G. Barto. *Reinforcement Learning: An Introduction*, Cambridge, UK: MIT Press, 1998.

[22] J. He, M. Ostendorf, X. D. He, J. S. Chen, J. F. Gao, L. H. Li, L. Deng. Deep reinforcement learning with a combinatorial action space for predicting popular Reddit threads. http://arxir.org/abs/1606.03667.

[23] D. Dowty. Compositionality as an empirical problem. *Direct Compositionality*, C. Barker, P. I. Jacobson, Eds., Oxford, UK: Oxford University Press, pp. 23–101, 2007.

[24] K. S. Tai, R. Socher, C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, pp. 1556–1566, 2015.

[25] S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, C. Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 1466–1477, 2016.

[26] R. Kaplan, C. Sauer, A. Sosa. Beating Atari with natural language guided reinforcement learning. *Computer Science*. http://adsabs.harvard.edu/abs/2017arXiv170405539K.

[27] F. Wu, Z. W. Xu, Y. Yang. An end-to-end approach to natural language object retrieval via context-aware deep reinforcement learning. http://arxir.org/abs/1703.07579.

[28] S. R. K. Branavan, H. Chen, L. S. Zettlemoyer, R. Barzilay. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, pp. 82–90, 2009. DOI: 10.3115/1687878.1687892.

[29] A. Pritzel, B. Uria, S. Srinivasan, A. Puigdomenech, O. Vinyals, D. Hassabis, D. Wierstra, C. Blundell. Neural episodic control. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, pp. 963–975, 2017.

[30] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017.

[31] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *Computer Science*. http://adsabs.harvard.edu/abs/2016arXiv161105397J.

[32] G. Lample, D. S. Chaplot. Playing FPS games with deep reinforcement learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, San Francisco, USA, pp. 2140–2146, 2017.

[33] Q. Y. Gu, I. Ishii. Review of some advances and applications in real-time high-speed vision: our views and experiences. *International Journal of Automation and Computing*, vol. 13, no. 4, pp. 305–318, 2016. DOI: 10.1007/s11633-

016-1024-0.

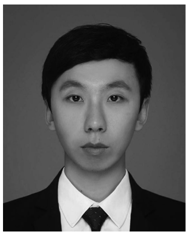
[34] S. Miyashita, X. Y. Lian, X. Zeng, T. Matsubara, K. Uehara. Developing game AI agent behaving like human by mixing reinforcement learning and supervised learning. In *Proceedings of the 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, IEEE, Kanazawa, Japan, pp. 489–494, 2017. DOI: 10.1109/SNPD.2017.8022767.

[35] Y. K. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, F. F. Li, A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Singapore, pp. 3357–3364, 2017. DOI: 10.1109/ICRA.2017.7989381.

[36] Q. V. Le. Building high-level features using large scale unsupervised learning. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Vancouver, Canada, pp. 8595–8598, 2013. DOI: 10.1109/ICASSP.2013.6639343.

[37] R. S. Sutton, D. McAllester, S. Singh, Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of Advances in Neural Information Processing Systems*, Denver, USA, pp. 1057–1063, 2000.

[38] D. R. Liu, H. L. Li, D. Wang. Feature selection and feature learning for high-dimensional batch reinforcement learning: A survey. *International Journal of Automation and Computing*, vol. 12, no. 3, pp. 229–242, 2015. DOI: 10.1007/s11633-015-0893-y.



Meng-Yang Zhang received the B. Sc. and M. Sc. degrees in automation from Qingdao University of Technology, China in 2012 and 2014, respectively. He is currently a Ph. D. degree candidate in control theory and control engineering at Shandong University, China.

His research interests include intelligent space technology and service robot, reinforcement learning, and knowledge construction based on ontology.

E-mail: zhangmengyang007@163.com

ORCID iD: 0000-0003-4267-1761

Guo-Hui Tian received the B. Sc. degree from Department of Mathematics, Shandong University, China in 1990, the M. Sc. degree in automation from Department of Automation, Shandong University of Technology, China in 1993, and the Ph. D. degree in automatic control theory and application from School of Automation, Northeastern University, China in 1997. He studied as a post-doctoral researcher in School of Mechanical Engineering, Shandong University from 1999 to 2001, and worked as a visiting professor in Graduate School of Engineering, Tokyo University of Japan from 2003 to 2005. He was a lecturer from 1997 to 1998 and an associate professor from 1998 to 2002 in Shandong University. At present, he is the professor in School of Control Science and Engineering, Shandong University, China. And also he is the vice director of the Intelligence Robot Specialized Committee of Chinese Association for Artificial Intelligence, the vice director of the Intelligent Manufacturing System Specialized Committee of Chinese Association for Automation, and the member of the IEEE Robotics and Automation Society.

His research interests include service robot, intelligent space, cloud robotics and brain-inspired intelligent robotics.

E-mail: g.h.tian@sdu.edu.cn (Corresponding author)

ORCID iD: 0000-0001-8332-3064

Ci-Ci Li received the B. Sc. degree in automation from the Northeastern University, China in 2014. She is currently the Ph. D. degree candidate in control science and engineering at Shandong University, China.

Her research interests include home service robot and object cognition.

E-mail: 201413043@mail.sdu.edu.cn

Jing Gong received the B. Sc. degree in automation from the Zhengzhou University, China in 2015. He is currently a master student in control science and engineering at Shandong University, China.

His research interests include home service robot, natural language processing and cloud robot system.

E-mail: gongjing689@gmail.com