Natural Spatial Language Generation for Indoor Robot

Zhiyu Huo Marjorie Skubic

University of Missouri-Columbia

***Abstract*—This paper proposes a spatial language generation system to find short, accurate and human-like descriptions for robots to communicate with a human user about the location of an object. The research focuses on building static spatial descriptions which use reference objects and directions to describe spatial relations. The system generates a natural spatial description in three steps. In the first step, it collects the sensory information and robot state to extract an environment model. Then, it builds a grounding model that describes the location of the target object, based on landmarks in the scene. After that it will generate the natural language description by imitating a human's talking style. A corpus of 149 spatial language commands for an indoor environment fetch task is used to train the system. An early-stage experiment was conducted and the results illustrate good potential for further development.**

Keywords-spatial language; language generation; robotics

I. INTRODUCTION

The interest in how a robot can be of assistance in our daily life continues to grow. For the robots working on household tasks, there is an increasing need for the capability to interact with human users; the interaction using spatial language is getting more attention from researchers. For robots that can interact with humans using spatial language, there are two complimentary robot challenges in a home-like environment. One is understanding natural language directives. For example, a human user directs a robot to fetch a target object by giving a spatial command. Another is spatial language generation, which lets a robot answer to a human user with the location of a target object by using natural spatial language. This paper focuses on the second challenge by building a language generation system for indoor robots.

Figure 1 shows an example of the spatial language generation task performed by a robot in an indoor environment. The human user is standing in the hallway between the living room and the bedroom, and he wants the robot to find the mug and tell him the location of it so that he can easily go right to it when he needs it. In this scenario, the human user expects the robot to give a description like "*Walk into the living room, then turn right and move forward, you will see the mug on the table*," or "*The mug is on the table in front of the couch in the living room*" which is a natural and friendly way of assisting and provides enough information to assure successful retrieval. Here, we focus on the generation of static spatial language which is the second example sentence above. The concept of static spatial language has been introduced in [1]. A spatial description of this type uses objects as references to describe a target location, i.e., "*behind the couch*" or "*on the table next to the bed*". The language generation task for indoor robots uses the sensory information collected from the environment to generate the static spatial language description. The generated description includes the spatial information in a large area so that it may be long and may have a complex structure which will make it difficult to be generated by a language template. This makes it different from other work on robot language generation and makes it a more challenging task. However, this kind of spatial language is human-like and provides more intuitive navigation information for a human user, particularly an elderly user.

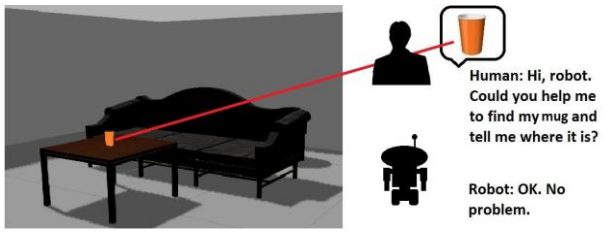


Figure 1. The scenario of an object searching and language generation task performed by a robot in a home environment. The Human said: "Hi, robot. Could you help me to find my mug and tell me where it is?" The robot answerd: "OK. No problem". Then the robot left to search for the mug.

There has been some significant work on the language generation. Reiter and Dale systemically described the approach to generate natural language with a probabilistic system [2]. Chen and Mooney presented a novel algorithm, Iterative Generation Strategy Learning (IGSL), for deciding which events to comment on in a soccer game [3]. The work in [4] introduced a novel model to generate spatial language. Angeli, et al proposed a multi-layer system generating natural language by two steps: content selection and surface realization [5]. Our work is the generation of spatial language

for robots in an indoor environment which is a different task. However, all of the work faces the same problem, which is generating human-like language using raw and unabstracted data. In the related work by Angeli et al., the process of language generation is split into two steps: the first one is content selection which selects the information to present from the raw data; and the second one is surface realization which infers the natural language from the selected content.

To enable the robot to provide easily understood spatial descriptions to a human user, we designed a multi-step system that follows the two steps mentioned above. The system first models the content of groundings from the sensory information collected in the environment, and then generates natural language from this intermediate result.

II. METHODOLOGY

A. The Multi-Layer Model of Spatial Description

The language generation system is based on our previous work on modeling spatial language and understanding spatial language directives, which has been developed to be a multi-layer system [6]. This system represents a natural spatial language description using four layers (Figure 2). The first layer is the natural language command. In the second layer, the words in the natural language description are grouped into chunks with meaningful tags by using part-of-speech algorithms [7]. In this layer, the words containing spatial information are detected and tagged, and the natural language is converted to a tree structure. In the third layer, the tree structure is translated into a grounding model in the form of the reference-direction-target (RDT) format presented in our previous work [1][6]. The RDT model is a standard representation with the information of landmark and spatial relation. In the RDT model, reference refers to an object that is used as a landmark reference to describe the location of another object. Direction represents the position relationship between objects, e.g., in front or to the left. It tells the robot where to search for the target. Target indicates the target furniture or target object being sought by the robot. It

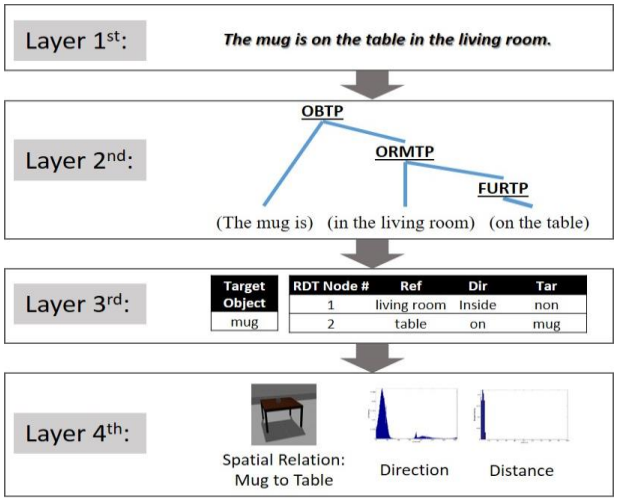


Figure 2. The multi-layer model of spatial language in our system

minimizes the uncertainty and ambiguity in human language and conveys a robot understandable message to let it seek the destination. Given a long spatial description with a complex structure, the chunks in the tree structure can be converted to RDT nodes, which describe a sequential action list or reference-based descriptions allowing the robot to move to find the target object. The fourth layer is a numerical representation of the spatial relations of both direction and distance between the objects in the environment. The data of this layer will be taken into the robot behavior model to infer the destination of RDT node.

B. System Overview

The goal of this work is to generate a natural language description of the position of a target object. For the example shown in Figure 1, the expected corresponding description is "*The mug is on the table in front of the couch in the living room*". The spatial description contains information about the environment. To deliver the position of the target object to a human user correctly, the robot should detect the environment and extract the spatial information that can best describe the position and then present them in natural language terms. In such a task, we let ε denote the information of the environment, and p denote the location and the orientation of the human user. Consider an objective function $h(\varphi,p,\varepsilon)$ of the natural description φ. The robot will search for a spatial description φ' with the largest function value:

$$\varphi' = argmax_{\varphi} h(\varphi, p, \varepsilon) \qquad (1)$$

The objective function determines the policies to select a spatial description which should: a) have accurate information for the human user to reach the target object, b) match the human spatial language syntax and human's language style and c) use the fewest number of words. However, to directly train the cost function by samples of φ, p and ε is a problem of great complexity. Here, we propose a multi-step process that splits the workflow into three steps:

(1) Model the Environment: the robot will build an environmental model which includes all the detected objects in its working environment until it finds the target object. All the objects in the environment are recorded. The information about an object is described by an Entity model that includes a category name, a coordinate vector, an orientation value and a unique ID of the object.

(2) Content Selection: Content selection is to decide what to say in a spatial description [2]. In our system, the content is represented by the RDT format grounding model presented in our previous work of spatial language grounding. In spatial language grounding, the RDT model is a result of inference from natural language. Here, the RDT model is built from the environment model and is a reverse procedure of the inference to robot destination.

(3) Surface Realization: Surface Realization determines how to convert spatial information into natural language [2]. After getting the RDT grounding model, the system generates natural language using a model trained by a

149-sentence template corpus which has been extracted from the CSISL spatial language corpus introduced in [8]. The CSISL contains 1024 indoor spatial descriptions collected from human volunteers, and the 149-sentence template represents all of the different types of language structures that were captured by the 1024 participant descriptions. The surface realization model takes the RDT model as input to select words and phrases to construct a human-like natural language sentence.

C. Build Envirionment Model

The first step to generate a static spatial language description is to build an environment model which is created by the robot perception system. Our system uses a depth camera as the robot sensor. With prior internal knowledge about the objects in the working environment, the robot can recognize the objects and capture their geometric features such as size, shape and orientation. The information of each object is integrated into a standard description named an Entity model. The Entity model is used to represent semantic objects handled in human spatial language. An Entity has: (1): an ID; (2): a name; (3): a coordinate vector; and (4): an orientation. The ID is the unique identification of an object in a robot task. The ID number of an entity is given by the sequence of detection. The name is the category of the object.

Figure 3. An example of using entity model to describe a chair. LEFT: the chair sample (The arrow illustrates its direction); MIDDLE: the 3-D point cloud of the chair; RIGHT: the 2-D point cluster ρ. The direction angle is $4\pi/7$rad (or 315°). The entity model $e=\{\text{“chair”},\rho,7/4\pi\}$.

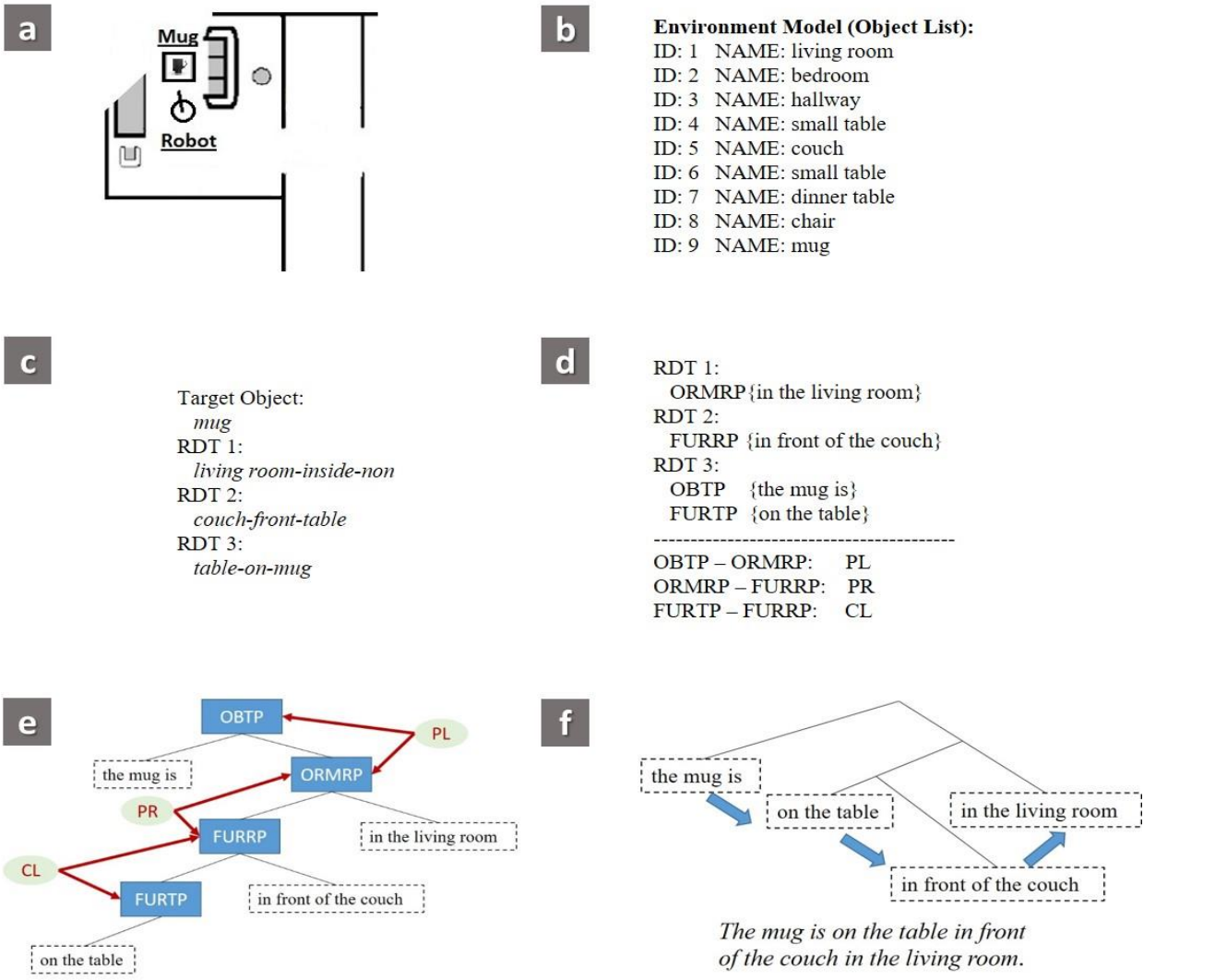


Figure 4. The procedure of natural spatial language generation. (a) The scenario when the robot detects the target object – mug. (b) The environment model. (c) The result of content selection. (d) The chunks used to infer the RDT nodes and their relations. (e) The tree structure built from the chunks and their relations. (f) The natural language description generated, the result of surface realization.

The coordinate vector is a 2D point cloud representing the object's projection on the floor. To reduce the computation and noise we down-sample the raw point cloud to the positions of cells in a grid map. The orientation of an entity is defined as the direction value of its functional front side in the ego-centric reference, e.g., a chair has its functional front as the direction that a person faces when sitting on it. The example of a chair entity is shown in Figure 3.

During a language generation task, the robot will keep building the environment model when seeking the target object in the working space until it finds the target. Thus it can build the environment model as a set of N entities $\varepsilon=\{e_1,...,e_N\}$ in the working environment. Figure 4(a) shows the scene for an object seeking task and Figure 4(b) the environment built from it.

D. Content Selection

Next, the robot generates an RDT grounding model with several RDT nodes from the environment model. The entities list ε generated from the last step is used to build a spatial relation list $\Gamma(\varepsilon)=\{\gamma_1,...,\gamma_M\}$. The list Γ includes M combinations between any two entities. For each combination, we use $\gamma_m=\{F_{direction}(e_a,e_b),F_{distance}(e_a,e_b)\}$ to represent two histogram vectors of direction and distance as the features of a spatial relationship. The spatial relation list $\Gamma(\varepsilon)$ is also called the world state (WS), which describes the spatial relations in the environment. The WS is then used to calculate the probability $P(y|\Gamma)$ of each possible RDT node y which will be used later in the objective function. In the spatial language grounding system, $P(y|\Gamma)$ is used to infer the destination that the robot should move to the RDT node y. Here the robot is considered as an entity e_{robot} which is equivalent to other object entities in Γ. Since the positions of the other entities are fixed, the inference to the destination is to adjust the robot to a pose where the e_{robot} for the Γ can maximize the probability $P(y|\Gamma)$. In the language generation system, e_{robot} is set by the pose where the robot finds the target and stops. The WS Γ is then built by e_{robot} and other entities detected in the environment.

To seek the best solution over all RDT nodes, an objective function is proposed. Let $\{y_1,...,y_K\}$ denote K RDT nodes that can be extracted from the environment (K is smaller than the number of all the possible RDT types). The decision on whether to select an RDT node is represented by a binary weight value w_k. The w_k is 1 when the RDT node y_k is selected to generate the spatial language description and is 0 if not selected. A number v_{k1k2} is a value between 0 and 1 which is the conditional probability $P(w_{k1}|w_{k2})$ for the selection of the two RDT nodes y_{k1} and y_{k2}. This value is learned from the RDT nodes extracted from the 149-sentence template corpus. Since the Γ is fixed in this step, we let $P_{yk}=P_\Gamma(y_k)$ which is the probability of y_k in the environment. Then we can compose the following objective function for the combination of all the K RDT nodes which is:

$$O(W)=\frac{\sum_{k=1}^{K}w_k P_{y_k}}{\sum_{k=1}^{K}w_k}+\sum_{\{k1,k2\}\in KK}v_{k1k2}w_{y_{k1}}w_{y_{k2}}P_{y_{k1}}P_{y_{k2}}-\alpha\frac{\sum_{k=1}^{K}w_k}{K} \quad (2)$$

The $W=[w_1,...,w_K]$ is a vector which includes all the w_k values. $KK=\{(1,2),(1,3),(2,3),...,(K-1,K)\}$ denotes a set of combinations of any two different numbers in vector [1,...,K]. The three parts in $O(W)$ represent different restrictions on the content to select. The first part encourages high probability groundings and the second part encourages the appearance of two related groundings that work together in the spatial description. The last part is used to get the shortest description. The constants $\alpha>0$ is adjusted by the training content data and we have $\alpha=0.1$ in our system. Here W is the only variable to be sought in the objective function. To get the best RDT model, we will infer a solution W' to maximize the objective function $O(W)$ which is:

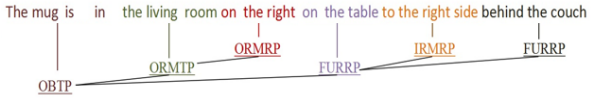
$$W'=argmax_W O(W) \quad (3)$$

The pose of human addressee is another restriction on the content to select. For example, when the robot and the person are in the same room, there is no need to present the information of room in the content. This restriction will work as a filter to remove some content.

Figure 4(c) shows the result of content selection from the environment model in Figure 4(b).

E. Surface Realization

After inferring the best RDT model, the last step is the transition from the RDT nodes to the natural language description presenting the location of the target object. Considering the diversity and uncertainty of human-like spatial language, it is difficult to use a fixed prototype framework on language generation. Inspired by our previous work in [9], we consider the output natural language description as a tree structure constructed by several clauses. An example of a tree-structured description is shown in Figure 5, which shows a language model grouping words into chunks (word phrases). Each chunk $c=\{\tau,\eta\}$ consists of a clause of text τ and a chunk type η. The chunk types and explanations are also shown in Figure 5. Thus the surface realization is to construct a tree structure with all the chunks placed in the best places. The tree structure is inferred by a probabilistic model counting the text and the relations between chunks. The potential relations that chunk A can have to chunk B include six possibilities: *neighbor-left(NL)*, *neighbor-right(NR)*, *parent-left(PL)*, *parent-right(PR)*, *child-left(CL)*, *child-right(CR)* (Shown in Figure 6). Assuming we have already inferred the best grounding model, which



Outside Room Target Phrase	ORMTP
Outside Room Reference Phrase	ORMRP
Object Target Phrase	OBTP
Furniture Target Phrase	FURTP
Furniture Reference Phrase	FURRP
Inside Room Reference Phrase	IRMRP

Figure 5. A chunking tree structure of a spatial description and the explanation of the chunk types.

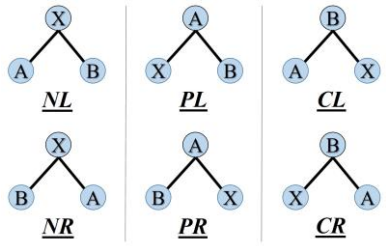


Figure 6. The six examples of the six possible relations between two chunks. For an instance, *NL*, the corresponding example demonstrates that how *A* is to be the *neighbor-left* to *B*.

includes an RDT chain $y=\{y_1,\dots y_J\}$ including J ($J \le K$, K is the number of possible RDT node before content selection) RDT nodes. Let let $\upsilon_{\eta A \eta B} \in \{NL,NR,PL,PR,CL,CR\}$ denotes the relation between two chunks with the names ηA and ηB where $\upsilon_{\eta A \eta B} \in \{NL,NR,PL,PR,CL,CR\}$. $\Upsilon = \{\upsilon_{\eta 1 \eta 2},\dots,\upsilon_{\eta J\text{-}1 \eta J}\}$ is the set of all the relations. Then the language generation work is to determine the Υ which can maximize the probability $P(\Upsilon)$ to generate a tree structure, which can be written as:

$$P(\Upsilon) = P\left(\{(\tau_1,\eta_1),y_1\},\dots\{(\tau_J,\eta_J),y_J\},\Upsilon\right) \propto \prod_{j=1}^{J} P(\tau_j,\eta_j|y_j) \prod_{j1=1}^{J} \prod_{j2=1}^{J} P\left(\upsilon_{\eta_{j1}\eta_{j2}},|\eta_{j1}\eta_{j2}, j1 \neq j2\right) \quad (4)$$

The conditional distribution can be trained by the 149 template descriptions that were derived directly from CSISL corpus collected from older adults. Figure 4(d) shows the chunks of the RDT nodes extracted in content selection and best matched relations of the chunks. Assume we extract P relations $Y=\{\upsilon_1,\dots,\upsilon_P\}$ between the chunks in this step.

After obtaining the chunks and their relations, the system then uses them to construct the tree structure. We use the chunk of the target object as the root of the tree. Two pools are created. Pool *A* contains the chunks not assigned to the tree and pool *B* contains the relations. An iterative algorithm is run on pool *B*, which places the chunks in pool *A* to the tree by the relations it involves in pool *B*. Then it removes the chunks from pool *A* and removes the relation in pool *B* (Algorithm 1). The iteration ends when pool *A* is empty or the iteration limit is reached. Figure 4(e) shows the tree structure generated from the chunks and relations presented in Figure 4(d). After building the tree, the system will generate the natural spatial language description using the in-order traversal of the tree [10].

The result (Figure 4(f)) is determined not only based on the words and tag of each grounding unit but also on their relations of nesting and ordering. This enables the system to mimic a human-like style in spatial language descriptions.

Algorithm 1

init: $A=\{c_1,\dots,c_J\}$, $B=\{\upsilon_1,\dots,\upsilon_P\}$, $t=1$, $TREE.ROOT=c_{obj}$
while $t<T$ **and** isempty(A) **is false**:
  for each υ in B:
    c_x,c_y=get_two_involved_chunks(υ)
    if ifintree(c_x) **xor** ifintree(c_y) **is true**:
      move $c_{notintree}$ to *TREE* by υ
      remove $c_{notintree}$ in A
      remove υ_p in B
    endif
  endfor
endwhile

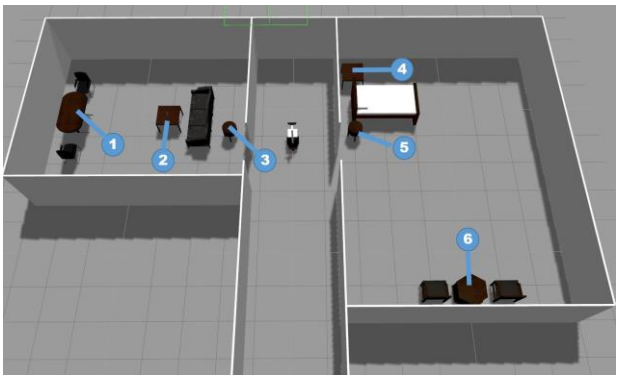
III. EXPERIMENT

To evaluate the system, an experiment will be performed first in a simulated indoor environment which includes a bedroom, a living room and a hallway between them (Figure 7(a)). Both rooms have relevant furniture pieces. This setting has been used in our previous work on spatial language grounding and matches our physical lab space [6]. The simulation environment is built using Gazebo3D platform [11]. The perception data and the control function of the robot were programmed the same as the version working in the physical environment so that system can also be migrated to the real world environment.

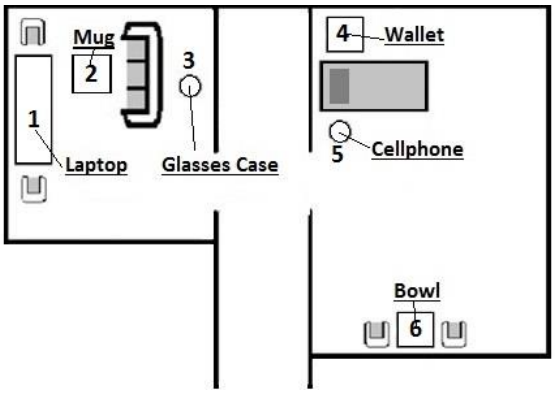
TABLE I The results of the early-stage experiment which includes six language generation tasks.

#	Object Target	RDT nodes	Natural Language Description
1	Laptop	*living room-inside-non table-beside-chair table-on-laptop*	*The laptop is on the table in the living room beside chairs.*
2	Mug	*living room-inside-non couch-front-table table-on-mug*	*There is the mug in the living room on the table in front of the couch.*
3	Glasses Case	*living room-inside-non couch-behind-table table-on-glasses case*	*The glasses case is in the living room on the table to the back of the couch.*
4	Wallet	*bedroom-inside-non bed-left-table table-on-wallet*	*The wallet is in the bedroom on the table to the left of the bed.*
5	Cellphone	*bedroom-inside-non bed-right-table table-on-cellphone*	*The cellphone is on the table in the bedroom to the right of the bed.*
6	Bowl	*bedroom-inside-non chair-beside-table room-right-non table-on-bowl*	*The bowl is on the table in the bedroom beside chairs to the far right wall.*

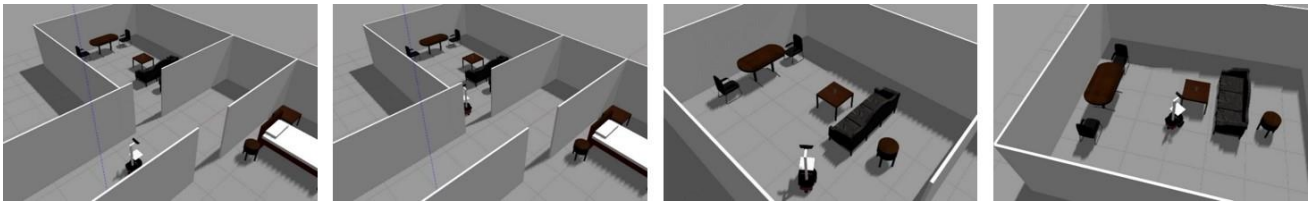
In a language generation test, the robot is initially positioned in the middle of the hallway and then starts to search for a target object after it receives the object name from the human user. It will keep on roaming in the working environment and builds the environment model until it finds the target. The target object can be placed in one of six different locations (Figure 7(b)). For each location, a static natural spatial language description will be generated. Here we list the results of an early-stage experiment in TABLE I which includes six descriptions generated by the robot. Although there are several metrics to score the performance of language generation, e.g., F-1 [12] and BLEU [13], which compare the similarity between the results and the ground truth, the best approach to assess a language generation result is to have it scored by a human. To give a more reliable assessment to our system, we will employ volunteer test subjects to score the spatial descriptions that are generated by the robot.



(a)



(b)



(c)

Figure 7. The 3D simulation scene built in Gazebo; The numbers label the furniture items where the target objects are placed on during the experiment. (b) The 2D floor plan of the scene for the experiment. (c) The object seeking procedure (from LEFT to RIGHT).

IV. CONCLUSION

The development of this blueprint was an effort to achieve a natural spatial language generation system. Our preliminary work addresses some of the challenges. The results of the early experiments confirm a decision to not use language templates but rather to use a human spatial language corpus to program a language generator.

This system is trained by the 149-sentence template corpus and tested by six cases in the same scene where the corpus was collected. There are two limitations of the current experiment. First, the number of test cases is too small. Additionally, since the test scene is the same as the scene used to train the language model, it is not enough to validate the system's suitability to other environments. In the future, the number of the scenes for testing will be increased and the furniture placement will be alternated. Even the results present accurate and human understandable descriptions, the language has a lack of variety. We will also compare our approach with other machine learning methods like inverse reinforcement learning and recurrent neural network. To improve this aspect, we will also test additional features and train the system by other corpuses.

ACKNOWLEDGMENT

This work was funded by the NSF under grant IIS-1017097.

REFERENCES

[1] Skubic, Marjorie, Zhiyu Huo, Tatiana Alexenko, Laura Carlson, and Jason Miller. "Testing an assistive fetch robot with spatial language from older and younger adults." In *RO-MAN, 2013 IEEE*, pp. 697-702. IEEE, 2013.

[2] Reiter, Ehud, Robert Dale, and Zhiwei Feng. *Building natural language generation systems*. Vol. 33. Cambridge: Cambridge university press, 2000.

[3] Chen, David L., and Raymond J. Mooney. "Learning to sportscast: a test of grounded language acquisition."*Proceedings of the 25th international conference on Machine learning*. ACM, 2008.

[4] Tse, Rina, and Mark Campbell. "Human-robot information sharing with structured language generation from probabilistic beliefs." Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE, 2015.

[5] Angeli, Gabor, Percy Liang, and Dan Klein. "A simple domain-independent probabilistic approach to generation."*Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010.

[6] Huo, Zhiyu, Tatiana Alexenko, and Marjorie Skubic. "Using spatial language to drive a robot for an indoor environment fetch task." In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 1361-1366. IEEE, 2014.

[7] Brill, Eric. "A simple rule-based part of speech tagger." In *Proceedings of the workshop on Speech and Natural Language*, pp. 112-116. Association for Computational Linguistics, 1992.

[8] Carlson, Laura, Marjorie Skubic, Jared Miller, Zhiyu Huo, and Tatiana Alexenko. "Strategies for Human-Driven Robot Comprehension of Spatial Descriptions by Older Adults in a Robot Fetch Task." *Topics in cognitive science* 6, no. 3 (2014): 513-533.

[9] Alexenko, Tatiana, Marjorie Skubic, and Zhiyu Huo. "Spatial Language Processing for Assistive Robots with" Deep" Chunking and Semantic Grammars." *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*. 2014.

[10] Schoenmakers, Berry. "Inorder traversal of a binary heap and its inversion in optimal time and space." *Mathematics of Program Construction*. Springer Berlin Heidelberg, 1992.

[11] Koenig, N., and A. Howard. "Gazebo-3D multiple robot simulator with dynamics (2003)." URL: http://gazebosim. org 3 (2013).

[12] Powers, David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).

[13] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation." In Proceedings of the 40th annual meeting on association for computational linguistics, pp. 311-318. Association for Computational Linguistics, 2002.