

# Parsing Parking Instructions for Self-driving Cars into Spatial Semantic Descriptions

Akari Inago<sup>1\*</sup>, Hiroshi Tsukahara<sup>2</sup>, Ichiro Kobayashi<sup>1</sup>

<sup>1</sup> Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan.

<sup>2</sup> Denso IT Laboratory, Inc., 2-15-1 Shibuya, Shibuya-ku, Tokyo 150-0002, Japan.

\* Corresponding author. Tel.: +81-80-1089-9145; email: g1320504@is.ocha.ac.jp

Manuscript submitted March 5, 2019; accepted May 15, 2019.

doi: 10.17706/jcp.14.5.328-338

---

**Abstract:** This study is motivated by an attempt to develop a system that can control a self-driving car by words. To achieve this, the verbal instructions we provide to the car must be grounded onto the real-world phenomena. With this background, in this study, we extend the framework of semantic description clause (SDC) proposed by Kollar *et al.* (2010) by adding two new semantic categories, VIEW and STATE, so as to be able to ground more variety of the instructions for driving a car in real-world environment. We developed a pipeline of a CCG parser, a reranker of parse trees and a converter from parse trees into SDCs. The result of parsing the instructions with the extended CCG grammar shows 84.7% accuracy on 6,019 various Japanese driving instructions. We finally convert the parse trees into SDCs using conversion rules, resulting in 90.4% accuracy on 5,638 parse trees.

**Key words:** Semantic parsing, syntactic parsing, spatial description clauses, combinatory categorial grammar.

---

## 1. Introduction

We aim to develop a technology that enables to ground verbal instructions for driving a car onto phenomena or objects observed in the real world. In this study, we especially focus on developing a parser which converts verbal instructions into the spatial representation based on spatial description clause (SDC) proposed by Kollar *et al.* [1]. To build the parser, we define the grammar rules based on the combinatory categorial grammar (CCG), which enables us to flexibly represent the relationship between the different spatial semantic categories in the form of a parse tree. We furthermore extend the SDC framework by adding two new semantic categories. In addition, we use the result of the parser as the input to a reranker that selects the best parse from the set of parses for each sentence. Finally, we develop a converter to convert these trees into SDCs in accordance with the defined conversion rules. Although we dealt with Japanese instructions in this study, our rules can be available for other languages if their own CCG parsers are built by changing the CCG bank (dictionary) because they are based on the subtree structures of parse trees which describe spatial semantics.

## 2. Related Work

As extraction of the spatial semantics from natural language, Kordjamshidi *et al.* proposed the Spatial Role Labeling (SpRL) as the task of identifying and classifying the spatial arguments of the spatial expressions mentioned in a sentence [2]. Based on this task, Pustejovsky *et al.* proposed SpaceEval that is a

combined information extraction and classification task with the goal of identifying and categorizing spatial information [3].

On the other hand, some researchers pose the grounding problem as inference on a graphical model structured in accordance with the well-studied parsing structure of language [4]-[6]. Tellex *et al.* developed an algorithm which builds a probabilistic graphical model called  $G^3$  based on the SDC structure [7]. Howard *et al.* described a variant of the grounding graph for constraints [8], and further Chung *et al.* subsequently improved it upon [9]. More recently, Paul *et al.* proposed a method to deal with abstract spatial concepts in instructions [10]. Arkin *et al.* proposed a variant of a contemporary probabilistic graphical model for language understanding in a context [11].

Our approach builds on the basic ideas presented by Kollar *et al.* and Tellex *et al.* [1], [7]. We use SDC to construct a grounding graph based on an spatial structure. The problem in constructing an SDC structure is how labels of SDC categories are assigned to the content of natural language. Kollar *et al.* treated this issue as a sequence labeling problem and solved it using conditional random fields [1] - they did not take the dependency relation among all elements in a sentence. Tellex *et al.* transformed the result of syntactic analysis achieved by the Stanford parser into SDCs [7]. Although Kollar *et al.* and Tellex *et al.* dealt with the sentences with well-dependency structure, we aim to deal with more colloquial and diverse sentences. We therefore propose a CCG-based grammar with spatial semantic categories and used it to extract a correct spatial semantic dependency tree from more complicated input sentences, and then convert them to SDC. There are two reasons why we use CCG. One reason is to convert the semantic content of a text into a logical formula such as a lambda expression and to make an inference by using external knowledge for instructions. The other reason is that it is difficult to directly extract spatial meanings from sentences in Japanese because Japanese sentences have characteristics of omitting words in it and being changeable in word order. This characteristic of Japanese sentences makes it difficult to directly convert semantic meanings into SDCs.

### 3. Background

#### 3.1. Spatial Description Clause

SDC was proposed as a method for describing spatial semantic relationships [1]. It categorizes the components of a sentence through the use of spatial semantics. Each linguistic component of a sentence is assigned one of three fields, figure  $f$ , relation  $r$ , and landmark  $l$ . Figure  $f$  represents the subject of the sentence, landmark  $l$  represents the object of the instruction, and relation  $r$  represents either an action to take place or a spatial relationship that determines the geometric relationship between the figure  $f$  and landmark  $l$  fields. Furthermore, each SDC is tagged with a semantic type, such as EVENT, defined according to the categorization by Jackendoff [12]. Here, EVENT represents an action sequence performed in the real world, and OBJECT represents an object in the real world such as a car or a human being. PLACE represents the location, and PATH represents either a route or a direction. These tags are used to define different probabilistic models according to the hierarchical structure of SDCs. A linguistic instruction containing spatial expressions can be represented by an SDC tree. Leaf SDCs contain only the words of their figure field and its grounding. Internal SDCs always contain text in their relation fields and SDCs in the figure and the landmark fields. If internal SDCs have text in their relation field, the corresponding SDCs in the landmark fields are always required. Fig. 1 shows the SDC tree for the instruction, “赤いバスの隣に停めて (Park next to the red bus).” The leaf SDC in Fig. 1 is OBJ( $f$  = 赤いバス {red bus}), which contains 赤いバス in the figure field. The internal SDC in Fig. 1 is PLACE( $r$  = の隣に {next to}  $l$  = OBJ( $f$  = 赤いバス {red bus})), which contains の隣に in the relation field and OBJ in the landmark field.

EVENT<sub>1</sub>(r = 停めて {park}  
 l = PLACE<sub>2</sub>(r = の隣に {next to}  
 l = OBJ<sub>3</sub>(f = 赤いバス  
 {the red bus})))

Fig. 1. Parse tree using the CCG-based grammar and its SDC tree.

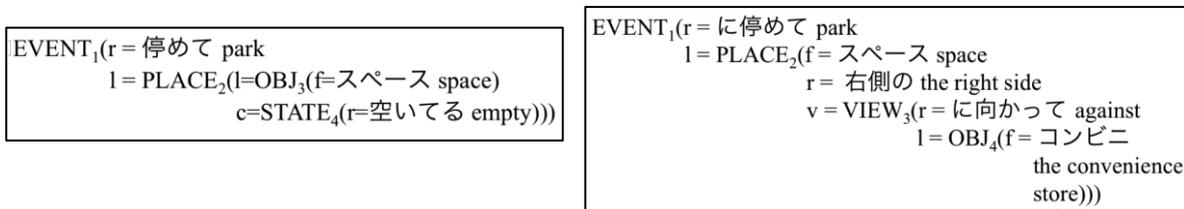
### 3.2. Combinatory Categorical Grammar

CCG is a lexicalized grammar formalism [13]. It contains two lexical categories, a base category, such as S, N, or NP, and a functional category, such as S\NP or NP/N. A functional category is defined recursively by combining a base category with two operators, “\” and “/”. CCG also incorporates combination rules, function application rules such as X/Y Y ⇒ X and Y X\Y ⇒ X. Here, X and Y are lexical category variables. In the following, we represent the application of such functional rules as (X ((X/Y) Y)) and (X (Y (X\Y))) to express the structure of parse trees in a simple form.

## 4. Conversion from Instructions to Parse Trees Using CCG

### 4.1. Extended SDC

In prior studies, Paul *et al.* and Arkin *et al.* focused on directly grounding natural language onto the phenomena and objects in the real world [10], [11]. We extend SDC to make more accurate for grounding natural language onto the real-world, and to accept various expressions in verbal instructions for parking a car. Tellex *et al.* used four types of semantic components: EVENT, OBJECT, PLACE, and PATH, in the framework for SDC [7]. We further add VIEW and STATE to these types. By adding these types, we can express one’s own viewpoint for parking a car and states of objects appearing in natural language, and therefore becomes able to more precisely reflect user’s intention. VIEW indicates the spatial direction from a base point to an another point, e.g., turn at the left corner, etc. VIEW has a role to disambiguate different expression about spatial relationship depending on viewpoint. For example, “left from north side” and “left from south side” is different. STATE indicates the state or the condition of an object appeared in parking instructions, e.g., the left side of a white car parking at slot 5 (state), the parking place not close to the toilet (condition), etc. Correspondingly, we add two fields, view *v* and condition *c* to each SDC nodes. View *v* represents a viewpoint used to describe the spatial relationship, and condition *c* represents the set of conditions that the SDC nodes are to satisfy. Fig. 2 shows examples of SDC trees for parking instructions that contain STATE and VIEW.



a. 空いてるスペースに停めて  
 (Park at the empty space.)

b. コンビニに向かって右側のスペースに停めて  
 (Park at the space on the right side against the convenience store.)

Fig. 2. SDC trees including STATE and VIEW.

### 4.2. CCG for SDC

In this study, following the SDC semantic categories, we use E (EVENT), O (OBJECT), L(PLACE), P (PATH), S (STATE), and V (VIEW) as base categories, and employ two function application rules. Although we use only two functional application rules, a number of complex functional categories can be built recursively

from atomic categories, hence sufficiently complicated syntactic trees are generated for treating problems in our target domain. Fig. 3 shows an example of parsing the instruction “赤い車の隣に停めて (Park next to the red car)” into a tree of spatial semantic categories using the grammar and an obtained SDC tree as the result. English translation is showed in the brackets.

The reason why we introduce CCG-based parsing method is that it makes possible to extract the uniform spatial semantic meaning regardless of the diversity of expressions in Japanese sentences, e.g., interchangeability of word order without changing the meaning. Fig. 4 shows an example of the CCG parse trees for two different expressions, 右側の空いているところに停めて and 空いているところの右側に停めて, of the same parking instruction, “Park at the space available on the right” in English, which can be transformed into the same SDC tree with the conversion rules described in Section 5.

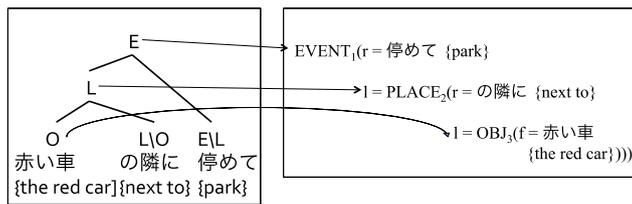


Fig. 3. Parse tree using the CCG-based grammar and its SDC tree.

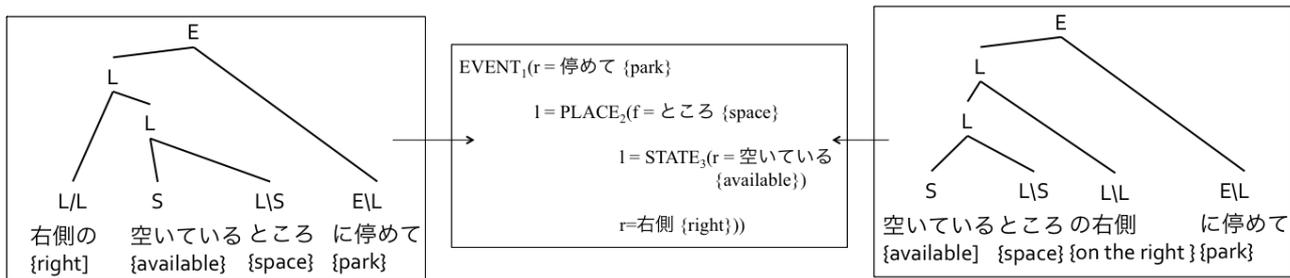


Fig. 4. Transforming different sentences with the same meaning into the same SDC tree.

### 4.3. CCG-Based Shift-Reduce Parsing

We chose a conventional shift-reduce parsing method in this study because our objective is not to propose a new parser algorithm but a new method to generate spatial descriptive clauses that represent more correct and precise spatial phenomena. In shift-reduce parsing, given an input sentence, our parser uses a stack of partial derivations, a queue of incoming words, and series of actions derived from the syntactic rules to build a derivation tree. The derivation tree can be transformed into SDC trees through a post-processing step. The set of action types used by the parser is {SHIFT, REDUCE-X}. Here, X represents a particular semantic category. The SHIFT action, following the standard definitions, just pushes the next incoming phrase onto the stack. The REDUCE-X action pops the top n nodes of the stack and combines them into a new node, which is pushed backed onto the stack. The category of the new node is X. The REDUCE-X action corresponds to a function application rule in the CCG grammar or else assigns the semantic category from the dictionary.

### 4.4. Score Model

The score of a candidate item is provided by the sum of the log likelihood of the sequential REDUCE-X actions. We use logistic regression to train the classifier for the SHIFT action and REDUCE-X actions. The number of REDUCE-X actions corresponds to the number of categories. Therefore, we use the

one-versus-rest method to perform multi-class classification. The features for a candidate are extracted when applying each REDUCE-X action. Table 1 shows the feature types used by the parser.  $S_i$  represents the  $i^{\text{th}}$  node on the top of the stack, and  $Q_i$  represents the  $i^{\text{th}}$  node on the top of the queue.  $p$ ,  $w$ , and  $c$  represent part-of-speech, word, and semantic category, respectively.

Table 1. Features Used by the Score Model

Description	Feature components
Unigram	$S_{0pw}, S_{0w}, S_{1pw}, S_{1c}, S_{1cw}, S_{2cw}, S_{3cw}, Q_{0pw}, Q_{1pw}, Q_{2pw}$
Bigram	$S_{0wS_{1w}}, S_{0wS_{1c}}, S_{0wQ_{0pw}}, S_{0wQ_{0p}}, S_{1cwQ_{0pw}}, S_{1cQ_{0pw}}, S_{1cwQ_{0p}}, S_{1cQ_{0p}}$
Trigram	$S_{0wS_{1c}Q_{0p}}, S_{0wS_{1cw}Q_{0p}}, S_{0wS_{1cw}Q_{0pw}}, S_{0pS_{1p}Q_{0p}}, S_{0wQ_{0p}Q_{1p}}, S_{0pQ_{0p}Q_{1p}}, S_{0wQ_{0pw}Q_{1p}}, S_{0pQ_{0p}Q_{1pw}}, S_{0wS_{1c}S_{2c}}, S_{0wS_{1cw}S_{2c}}, S_{0wS_{1c}S_{2cw}}, S_{0pS_{1p}S_{2p}}$

#### 4.5. Reranking

Referring to [14], we implemented a reranker that fixes the order of candidate parse trees and selects the highest one among the results. In our shift-reduce parser, we use only the local structure of a parse tree as features to parse input natural language, but we use the global structure of the tree in the reranker. Our reranker scores the plausibility of the structure of a parse tree in terms of a posterior probability estimated by binary logistic regression. – For preparing train data, we first generate candidates of parse trees by our parser. We give a positive label to a correct parse tree and negative labels to other incorrect trees of a candidate. Our reranker focuses on four types of features: (1) frequency of the patterns of functional application rules in parsing, (2) n-grams of words, (3) word-category pairs, and (4) structure of subtrees. (1) represents how often the patterns of the function application rules,  $X/Y\ Y \Rightarrow X$  or  $Y\ X\ Y \Rightarrow X$ , appeared in the parsing process. (2) represents unigrams, bigrams or trigrams of strings of words assigned categories in the parse tree. (3) represents a pair of a string of words and its category. (4) represents the plausibility of the subtrees, which is evaluated by the existence of the plausible subtrees in the parse tree. Plausible subtrees are trees in the correct answer data that include the leaves of the parse tree, whose depth is two through five. Table 2 shows an example of features extracted from “赤い車の隣に停めて (Park next to the red car).” Frequency corresponds to the frequency of the patterns of functional application rules. N-gram corresponds to n-grams of words. Pair corresponds to word-category pair, and Subtree corresponds to the structure of subtrees.

Table 2. Features Extracted from “赤い車の隣に停めて (Park Next to the Red Car)”

Description	Feature components
Frequency	$O_{(L\ O)}_1, L_{(E\ L)}_1$
N-gram	赤い車, の隣に, 停めて, 赤い車_の隣に, の隣に_停めて, 赤い車_の隣に_停めて
Pair	赤い車_O, の隣に_(L\ O), 停めて_(E\ L)
Subtree	$(E\ (L\ (O))), (E\ (L\ (L\ O))), (E\ (L\ (O\ (L\ O))))), (E\ (L\ (O))\ (E\ L)), (E\ (L\ (L\ O))\ (E\ L)), (E\ (L\ (O\ (L\ O)))\ (E\ L))$

## 5. Conversion from Parse Trees to SDCs

### 5.1. Conversion Rule

This section describes a method for converting a parse tree to an SDC tree. We use conversion rules based on heuristic knowledge to convert each subtree whose depth is one. There are 49 rules to meet these conditions. The conditions are roughly divided into six types: (i) which category the node has, (ii) whether the target node is the leaf of a tree, (iii) how deep structure the tree has, (iv) whether the parent node and the right child node have the same category, (v) whether the target child node has a word that indicates space, and (vi) whether the left node has textual information in the relation field of the SDC representation. Converting a parse tree into an SDC tree is summarized into the four steps below:

1. Choose a subtree whose depth is one.
2. Find its conversion rule as shown in Fig. 5.
3. Apply the selected conversion rule.
4. Repeat 1-3 until all subtrees are converted.

Fig. 5 shows an example of converting L ( O (赤い車 {the red car}) (L\O) (の隣に {next to})) to PLACE(r=の隣に {next to} l=OBJ(f=赤い車 {the red car})). This corresponds to Steps 2 and 3. In the figure, A is the node with the parent category and both B and C are the child nodes with words and the category or SDCs and the category. This conversion shows that subtrees (left-hand side of the arrow) are converted into SDCs (right-hand side of the arrow).  $N_w$  represents the word  $w$  that the node  $N$  has.  $N_c$  represents the category that the node  $N$  has.  $cat\_to\_SDC(C)$  represents the conversion of the base category  $C$  to an SDC.

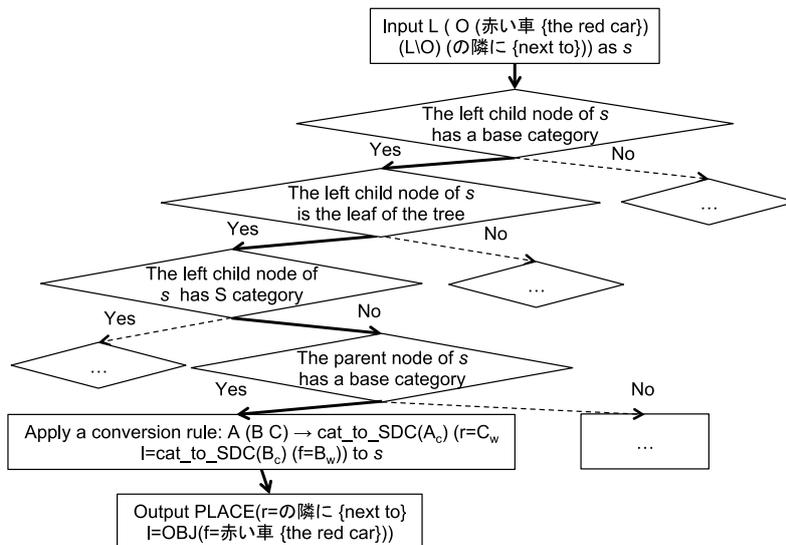


Fig. 5. Example of converting a subtree to a SDC.

### 5.2. Complementation for Omitted SDCs

Some instructions do not have enough information to ground onto phenomena or objects observed in the real world. Then we prepare conversion rules to compensate for necessary information in SDC trees. There are four types of SDCs to be complimented. The first type is the case in which the root SDC is PLACE. The second type is the case in which the root SDC is OBJECT. The third type is the case where PLACE has text in the relation field but no child SDCs in the landmark fields. The fourth type is the case in which PLACE has text in the relation and condition fields, but no child SDCs in the landmark fields. Table 3 shows a part of the complementation rules. \* can take arbitrary SDCs or texts for simplicity. The left-hand side of the

conversion rules shows imperfect SDCs, and the right-hand side of the rule is the result of complementation for those SDCs.

Table 3. Conversion Rules for the Supplement of Imperfect SDCs

Type	Conversion rule	Example
1	PLACE(*) → EVENT(r=停めて{park} l=PLACE(*))	PLACE(f=スペース{the space} r=前の{in front of} l=OBJ(f=歩行者{a walker})) → EVENT(r=停めて{park} l=PLACE(f=スペース{space} r=前の{in front of} l=OBJ(f=歩行者{a walker}))) The space in front of walker. → Park the space in front of walker.
2	STATE(*) → EVENT(r=停めて{park} l=PLACE(f=スペース{space} c=STATE(*)))	STATE(r=が空いてる{is available} l=PLACE(r=右端 {the far right of} l=OBJ(f=駐車場{the parking lot}))) → EVENT(r=停めて{park} l= PLACE(f=スペース{space} c=STATE(r=が空いてる{is available} l=PLACE(r=右端{the far right of} l=OBJ(f=駐車場{the parking lot})))))) The far right of the parking lot is available. → Park the available space that is the far right of the parking lot.
3	(*PLACE(r=*)*) → (*PLACE(r=* l=PLACE(f=スペース{space} c=STATE(r=空いてる {available})))*)	EVENT(r=停めて{park} l=PLACE(r=真ん中に{the midst of})) → EVENT(r=停めて{park} l=PLACE(r=真ん中に{the midst of} l=PLACE(f=スペース{space} c=STATE(r=空いてる{available})))))) Park the midst. → Park the midst of available space.
4	(*PLACE(r=* c=STATE(*))* → (*PLACE(r=* l=PLACE(f=スペース{space} c=STATE(*))*))	EVENT(r=停めて{park} l=PLACE(r=右側{the right side of} c=STATE(r=空いてる {available}))) → EVENT(r=停めて{park} l=PLACE(r=右側{the right side of} l=PLCAE(f=スペース{space} c=STATE(r=空いてる{available})))))) Park the available right side. →Park the right side of available space.

## 6. Experiments

We conducted experiments to evaluate our parser, reranker and SDC converter.

### 6.1. Experiment Settings

We built a corpus consisting of 6,019 Japanese parking instructions for evaluation. The parking instructions were collected by crowd-sourcing where 168 workers are given the picture of the parking scene (Fig. 6) and are asked to give instructions to park his/her car at the three candidate parking slots in front of a convenient store, utilizing either the spatial relation between their cars and the candidate slots or the spatial relation between a few landmarks and the candidate slots. Semantic categories were annotated manually by the first author and checked by the second author. In this study, we do not deal with instructions that contain PATH because there were few samples of this type in the collected corpus. Consequently, there are 14 categories not including S or V and 45 categories including all categories (Table 4). There are 381 parse tree patterns of car parking instructions, 92 types without STATE or VIEW, and 289 types with STATE or VIEW (Table 5). The parsing accuracy for each parse tree pattern was evaluated using 10-fold cross validation.

Table 4. Semantic Categories Found in Collected Parking Instructions

Category
L, (L\O), (L/L), (O/O), (E\L), (O/O)\O, (O/O)\L, O, (L/L)\O, (L/L)\L, (E\O), (L\L), S, (S/S), (L\S)\O, (L/L)\S, (S\L), (L\S)\S, (L\L)\S, (S/S)\O, V, (L\S)\V, (L\V)\O, (V\O), (V\L), (L\V), (L\V)/L, E, (L\S), (L\S)/L, L\ (S\L), (O/O)\S, (E\S)\L, (S\O), (O\S), (E\S), (O\V), (L\V)\L, (O/O)\V, (L/L)\V, (E\L)/S, (E/S)\L, (E/S), (L\L)\O, (S\L)\O



Fig. 6. Parking scene.

Table 5. A Part of the 381 Parse Tree Patterns Found in Collected Parking Instructions

	Structure of parse trees	Example	Num	Parsing Accuracy	Reranking Accuracy
1	(E (L (O (L\O)) (E\L)))	入り口 {gate} の近くに {near} 停めて {park} Park near the gate.	586	0.916	0.928
2	(L (O (L\O)))	コンビニ {the convenience store} 前 {in front of} In front of the convenience store.	512	0.902	0.902
...	...	...	...	...	...
38 0	(S (L ((L/L) L) (S\L)))	真ん中の {central} 駐車場 {a parking space} が {is} 空いています {available} The central parking space is available.	1	1.0	0.0
38 1	(E (L (S (O ((O/O) (O ((O/O) (O (O/O)\O) (O (O/O)\O) O) (S\O)) (L\S)) (E\L)))	黒い車 {the black car} と {and} 白い車 {the white car} の間の {between} 歩行者 {a pedestrian} がいる {is} ところ {the place} に停めて {park} Park the place where a pedestrian is between the black car and the white car.	1	0.0	0.0

## 6.2. Evaluation of Parser and Reranker

Table 4 shows the parse tree patterns based on the combination of semantic categories assigned by parsing the collected instructions. Num indicates the number of instructions whose derived trees have the structure depicted in the second column in the table. Parsing accuracy indicates the proportion of instructions correctly parsed by our parser. Reranking accuracy indicates the proportion of correct parse trees output by our reranker. From the parsing accuracy for each parse tree pattern, in many cases, parse tree patterns that have relatively simple structures or frequently appeared tend to get a high score.

On the other hand, the characteristics for the types with low accuracy are caused by the complicated structure of their parse trees or include complicated categories such as (L/L)\X, where X indicates any SDC category. Comparing the results before and after adopting the reranker, reranking accuracy for parse tree pattern is improved in most cases. In the case where reranking accuracy is much higher than parsing accuracy, we found that words tend to be split correctly. While, in the cases in which both parsing and reranking accuracies are low, we found that the shape of a parse tree tends to be incorrect, even if the words are correctly split. Table 5 shows the parsing and reranking accuracies for all of the experiments. As for accuracy, if the whole structure of a parse tree corresponds to that of the referential data, we regard it as correct. Ratio of successful parsing shows the proportion of sentences that a parse tree can be built. Accuracy of derived trees shows the proportion of derived trees that are correct. The low parsing accuracy is considered due to the insufficient accuracy of scoring candidate parse trees. It is because Japanese postposition articles have plural candidate semantic categories. When reducing the word with plural

categories, the number of candidates is exponentially increased. Moreover, Japanese tends to have various expressions for the meaning of one English word, for example, as for the word 'left', it can be meant by 左 (left), 左側, 左横, 左隣, etc (left side), 左手 (left hand), 左方向, 左方, 左の方, etc (leftward), etc, to name a few. Moreover, those Chinese characters can be expressed in "Hiragana", e.g., ひだり for 左. This causes many varieties of SDC trees and we need a lot of corpuses for high parsing accuracy.

Comparing the results of the parser with that of reranking, the accuracy of reranking and derived trees is improved. This is because the reranker uses information regarding the global structure of parse trees as features, whereas the parser uses information regarding the local structure.

### 6.3. Evaluation of the Accuracy of Conversion from Parse Trees to SDC

We evaluate use the 5,638 parse trees that are the outputs of our reranker using the data. We calculated the accuracy of our SDC converter, employing 10-fold cross validation for evaluation. Table 7 shows the accuracy of the converter's output SDC compared to the referential data. Accuracy from all trees shows the accuracy when the inputs of the converter are all of the outputs of the reranker. Accuracy from correct trees shows the accuracy when the inputs of the converter are only the correct outputs of the reranker. The numerical values with parentheses in Table 7 show the number of experimental data. This table indicates that all of the correct outputs of the reranker are correctly converted into SDCs. We found that the defined conversion rules are enough to convert parse trees into SDCs.

Table 6. Parsing and Reranking Accuracies

Method	Accuracy	Ratio of successful parsing	Accuracy of derived trees
Parsing	0.618	0.937	0.659
Reranking	0.847	0.937	0.904

Table 7. Conversion Accuracy

Accuracy from all trees	Accuracy from correct trees
0.904 (5638)	1.0 (5097)

## 7. Conclusion

In this study, we introduced CCG-based syntactic rules that enable us to represent the spatial semantic dependency relationship as a parse tree. We then proposed a method for transforming the parse tree into SDCs. We also extended the semantic types of SDC by adding STATE and VIEW. We developed a CCG-based shift-reduce parser and evaluated its performance. The accuracy was not initially high enough but has been improved significantly by our reranker, which uses information regarding global structure of parse trees as features. Furthermore, we developed a converter that converts parse trees into SDCs. This makes it possible to convert the parse trees into SDCs perfectly for the corpus we dealt with in this study. In future work, we will deal with various driving instructions. Furthermore, we aim to introduce a logical formula such as lambda expression to infer when recognizing the phenomena occurring in the world by using knowledge-based information.

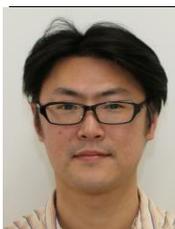
## References

- [1] Kollar, T., Tellex, S., Roy, D., & Roy, N. (2010). Toward understanding natural language directions. *Proceedings of 5th ACM/IEEE International Conference on Human-Robot Interaction*.

- [2] Kordjamshidi, P., Otterlo, M. V., & Moens, M. (2010). Spatial role labeling: Task definition and annotation scheme. *Proceedings of the 7th Conference on International Language Resources and Evaluation* (pp. 413-420).
- [3] Pustejovsky, J., Kordjamshidi, P., Moens, M., Levine, A., Dworman, S., & Yocum, Z. (2015). Semeval-2015 task 8: Spaceeval. *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 884-894).
- [4] Matuszek, C., Herbst, E., Zettlemoyer, L., & Fox, D. (2012). Learning to parse natural language commands to a robot control system. *Proceedings of the 13th International Symposium on Experimental Robotics* (pp. 403-415).
- [5] Boularias, A., Duvallet, F., Oh, J., & Stentz, A. (2015). Grounding spatial relations for outdoor robot navigation. *Proceedings of IEEE International Conference on Robotics and Automation*.
- [6] Oh, J. H., Suppé, A., Duvallet, F., Boularias, A., Navarro-Serment, L., & Hebert, M. (2015). Toward mobile robots reasoning like humans. *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (pp. 1371-1379).
- [7] Tellex, S., Kollar, K., Dickerson, S., Walter, M. R., Banerjee, A. G., & Teller, S. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. *Proceedings of the 25th AAAI Conference on Artificial Intelligence* (pp. 1507-1514).
- [8] Howard, T. M., Tellex, S., & Roy, N. (2014). Natural language planner interface for mobile manipulators. *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 6652-6659).
- [9] Chung, I., Propp, O., Walter, M. R., & Howard, T. M. (2015). On the performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5247-5252).
- [10] Paul, R., Arkin, J., Aksaray, D., Roy, N. & Howard, T. M. (2016). Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. *Proceedings of Robotics Science and Systems*.
- [11] Arkin, J., Walter, M. R., Boteanu, A., Napoli M. E., Biggie, H., & Kress-Gazit, H. (2017). Contextual awareness: Understanding monologic natural language instructions for autonomous robots. *Proceedings of 26th IEEE International Symposium on Robot and Human Interactive Communication*.
- [12] Jackendoff, R. (1983) *Semantics and Cognition*. Cambridge: MIT Press.
- [13] Steedman, M. (2000) *The Syntactic Process*. Cambridge: MIT Press.
- [14] Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 173-180).



**Akari Inago** is a master student at Ochanomizu University since 2017. She received the bachelor of computer science at the same university. Her research interest includes natural language processing and artificial intelligence.



**Hiroshi Tsukahara** is a senior researcher at Denso IT Laboratory, Inc. since 2006. He received the bachelor, master of physics and Ph.D. degrees from Chuo University in 1994, 1996, and 1999, respectively. He is working on the development of spoken dialogue interfaces for in-car information devices, self-driving cars and human communication robots.



**Ichiro Kobayashi** is a professor of computer science at Ochanomizu University. He was an associate professor at Hosei University from 1996 to 2003 and at Ochanomizu University from 2003 to 2010. He has been a visiting research scholar at National Institute of Advanced Industrial Science and Technology since 2017 (AIST). His main research interests are in the area of natural language processing and artificial intelligence.