

Query Generation for Resolving Ambiguity in User's Command for a Mobile Service Robot

Kazuho Morohashi and Jun Miura
Department of Computer Science and Engineering
Toyohashi University of Technology

Abstract—This paper describes a method of generating queries for resolving ambiguities in the user's command in service robotics applications. We deal with *bring-me* tasks, in which a robot brings a user-specified object from a distant place. A user's command may be ambiguous due to various reasons such as the uncertainty in his/her knowledge of the distant scene and the robot's knowledge. In such a case, the robot compares its recognition result with the command and generates a query for disambiguation. Based on previous VDQG (visual discriminative question generation) work, we develop a method for query generation using the concept of attribute contrast with the attribute categorization. We verified our method by comparing the user and the generated queries. We also implemented a robotic system, as a proof-of-concept, that can interact with the user and certainly achieve *bring-me* tasks.

I. INTRODUCTION

Lifestyle support is one of the promising applications of robotic technologies. As many countries are facing aging/aged society, service robots are needed that support people in their everyday life. One of the ubiquitous tasks at home is to bring a specific object from a different room.

A typical approach to achieving such a task is a combination of user's commands and the robot's autonomy. By specifying a name (or characteristics) of a target object, a robot moves around at a remote site, searches for the object, finds and picks it up, and takes it to the user. When the user's specification of the target object is complete, that is, a sufficient amount of information for identifying the object is given in the command, it is relatively easy for the robot to find it. Otherwise, the robot may have to obtain further information from the user to identify the target. Such an *ambiguous* situation often arises by several reasons such as an incomplete specification of the target object by the user, lack of sharing a common knowledge of the environment and objects, unexpectedly tricky cases (for example, almost the same but a different object exists near the target against the user's expectation). Such a situation is common even among people, and they solve ambiguities through appropriate interactions. We therefore would like robots to have an ability to solve the ambiguities in the user's commands through human-robot interaction (HRI).

When the user can see images of a remote scene, possibly through a camera on the robot, GUI-based interfaces operated by touch [1] or eye-gaze [2] are intuitive and useful. If this is not the case, that is, when the user cannot see such images nor operate such devices, a speech-based interface is useful.

This paper deals with HRI in such a situation.

In speech-based interaction, the robot needs to explain the situation at a remote scene by text. Various approaches to generate descriptive texts for images have been developed such as Image Captioning [3] and Visual Question Generation (VQG) [4]. The task in these approaches is to generate informative and/or natural descriptions and not to generate queries. Visual Discriminative Question Generation (VDQG) [5] generates questions, expected answers to which will be useful for discriminating one image from the other. This work provides good ideas for our query generation tasks but needs to be modified to fit our service robot scenarios.

This paper proposes a method of generating queries for resolving ambiguity in user's commands in *bring-me* tasks. Based on the recognition of candidate objects with attribute extraction, the robot identifies the useful attributes for resolving the current ambiguity most effectively. The contribution of this paper is twofold. One is to develop a method of query generation. The other is to evaluate it with both query generation quality and real robot experiments.

II. RELATED WORK

A. Dialog systems for robots

Research on dialog systems has actively been studied in the intelligent systems domain including robotics. Since the user's commands or orders often include ambiguities, how to resolve them is one of the important research issues. Makihara et al. [6] developed an object recognition system supported by dialog. To cope with the case where the system fails to recognize a target object correctly, it requests the user to provide additional information (e.g., a possible location of the target in the image).

Dialog systems are also used for *grounding* physical entity like an object or a specific space and/or resolving ambiguity/inconsistency between entities and symbols. Spexard et al. [7] dealt with a task of human-robot joint environment exploration and the robot generates questions when some inconsistency is found in grounding relationships (e.g., different names are given to a spatial entity). Deits et al. [8] developed a system that can interactively resolve ambiguities in user's commands. They use a *grounding graph* that probabilistically relates symbols with physical (recognized) objects, and the system generates three types of questions for yes-or-no, targeted, and reset, if necessary. In these methods, the robot and the user share a view of the environment.

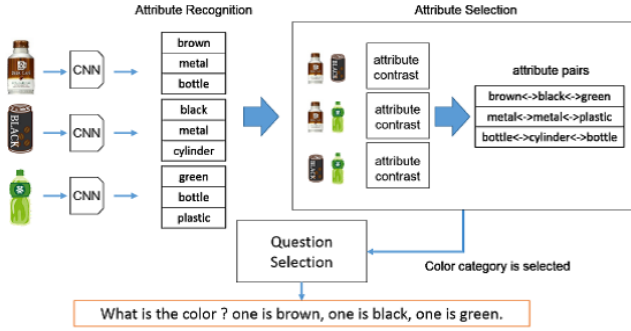


Fig. 1. Overview of the proposed system.

B. Visual-text multimodal processing

Image captioning [3], [9], [10] is a task of describing texts for a given image, such as “a man is sitting on a chair,” for the image of such a scene. In image captioning, not only object recognition but also recognizing what is going on an image are required. These techniques can be used for the robot to explain the situation of a remote site to the user. However, in our service task application, the description should focus on a part of the scene that is related to the ambiguity of the user’s command.

Visual Question Answering (VQA) [11], [12] is another multimodal (image and text) task, in which given a pair of an image and a question, an appropriate (correct) answer should be generated. A latent space of image and text is usually constructed and utilized to learn their relationships. Shih et al. [4] proposed to introduce an attention mechanism for generating a better answer to the query. VQA methods are not directly used for our current task but might be used for more bidirectional interaction between robots and human.

Visual Question Generation (VQG) [13] also deals with image and text, but try to generate *natural* questions for a given image. Li et al. [5] developed a system to generate a good question to distinguish a given pair of images. They call this task *Visual Discriminative Question Generation (VDQG)*. The system analyzes the images to get a set of object attributes which are useful for constructing a question. Although this work is to distinguish arbitrary (but similar) images and not intended for robotic applications, the idea of pursuing discriminative attributes is adopted in our system.

III. CATEGORY SELECTION AND QUERY GENERATION

A. Overview

Fig. 1 illustrates an overview of the proposed system. The robot has a basic ability of general object recognition like Yolo [14]. The result of recognition will be the input to the system. When the robot needs to choose one object among n objects, the input to the system is a set of n bounding boxes in the RGB image. Each object region is processed with a CNN-based attribute extractor to generate a ranked list of attributes. Then, those ranked lists are analyzed with attribute distinctiveness and attribute category matching. Finally, the

TABLE I
ATTRIBUTES AND ATTRIBUTE CATEGORIES.

category	attributes
appearance	bright, dark, light, new, old, wet, colorful, dirty, dry, plaid, beautiful, striped, painting, multi-colored, colored, clean
attachment	logo, no-logo, lid, no-lid, label, no-label, lettering, no-lettering, grip, no-grip
color	white, black, blue, green, red, brown, yellow, orange, gray, gold, silver, beige, cream, grey, pink, purple, blonde
shape	circle, oval, triangle, square, rectangle, quadrilateral, diamond, parallelogram, trapezoid, polygon, sphere, prism, cube, rectangular, pyramid, cylinder, cone, polyhedron, corner, side, plane, edge, flat, circular, wide, bottle, cup
material	brick, cement, concrete, glass, gravel, metal, plastic, sand, stone, wood, iron, steel, cloth
size	large, small, big, little, tall, short, high, low
state	bent, empty, many, stack

best category to discriminate all objects is selected and a query on that category is generated.

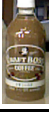


B. Attribute extraction

1) *Network structure*: We use Residual Network with 152 layers (ResNet-152) [15] pre-trained with ImageNet [16] for image feature extraction. The pre-trained ResNet outputs 1,000 class labels. We use the output of its layer just before full-connected ones as the image feature. This feature is the input to a three-layer MLP (multi-layer perceptron). The dimension of the input to the MLP is 2,048 and the number of units in the middle layer is 1,000. The dimension of the output is 95, which corresponds to the number of attributes we used.

2) *Attributes, dataset, and training*: We define 95 attributes, grouped into seven categories, as shown in Table I. The attributes are manually annotated to a set of real and simulated images of bounding boxes detected by Yolo. We prepared 17 images of drinks and 19 images of cups. These images are augmented by color conversion, adding noises, and reverting images. We finally have 476 and 665 images for drinks and cups, respectively. Table II shows example images and annotated attributes. Since an object usually has multiple labels, we trained the network for a multi-label classification problem. We randomly divided the images into training (80%), validation (10%), and test (10%) and trained the network until the validation accuracy exceeded 0.95. The averaged test accuracy by the trained network was about 0.90.

Note that the current sets of attributes and categories are manually generated considering applications not only to objects used in the paper but also to general objects. They are not exhaustive nor well organized. For example, the set of attributes includes ones like “cup” and “bottle.” These words could be used for describing attributes (juice in a cup or juice in a bottle, for example) but are usually used for indicating object classes and might have been rephrased as “cup-like” and “bottle-like.” The refinement of the sets will certainly be necessary.

TABLE II
EXAMPLE IMAGES AND ATTRIBUTES.

		
brown, white, multi-colored, label, plastic, lid, bottle, short	brown, paper, cup, no-grip, no-lid, short	brown, wood, cup, grip, no-lid, short

C. Criteria for Attribute selection

Queries for disambiguation are usually about object attributes, and the more unique an attribute is for a specific object, the better the object will be identified. Li et al. [5] proposed to use the following three criteria for choosing attributes for discriminating two images in VDQG tasks: *attribute score contrast*, *question similarity*, and *visual dissimilarity*. Considering the difference between their task and ours (bring-me task), we decided to use only attribute score contrast with the attribute categorization to generate a query on the most discriminative category. The attribute score contrast is defined as:

$$s(i, j) = v_i^A (1 - v_i^B) \cdot v_j^B (1 - v_j^A), \quad (1)$$

where i, j are a pair of attributes, A, B are image regions to compare, v_i^* is the attribute score. This value is large when two attributes appear strongly only in respective regions.

D. Category selection algorithm

We select the most effective category for query generation. Different from comparing a pair of images, in our remote object search tasks, the number of objects to discriminate is not fixed. We therefore calculate the attribute score contrast for every pair of objects and select the best category based on all attribute score contrast values.

The algorithm for category selection is as follows (see Fig. 2 for a three-object case):

- 1) Choose a pair of objects among all object candidates (see Fig. 2(a)).
- 2) Extract attribute scores for every pair of attributes in every category. Fig. 2(b) shows the case where “red” and “blue” attributes in “color” category are selected.
- 3) Calculate the score for each attribute pair (see Fig. 2(c)) and record the best scored pair for each category.
- 4) Select the category that has the highest averaged best score (see Fig. 2(d)).

E. Query generation

Our query generation is straightforward once the category to ask is determined. So we prepare a fixed query for each category. For example, we use the phrase “What is the color?” if the selected category is “color.” To enable the user to answer the query in a more informative way, descriptions of that category for all candidates objects are also added. In the case of Fig. 2(a), the final query becomes: “What is the color? One is red, one is blue, and one is green.”

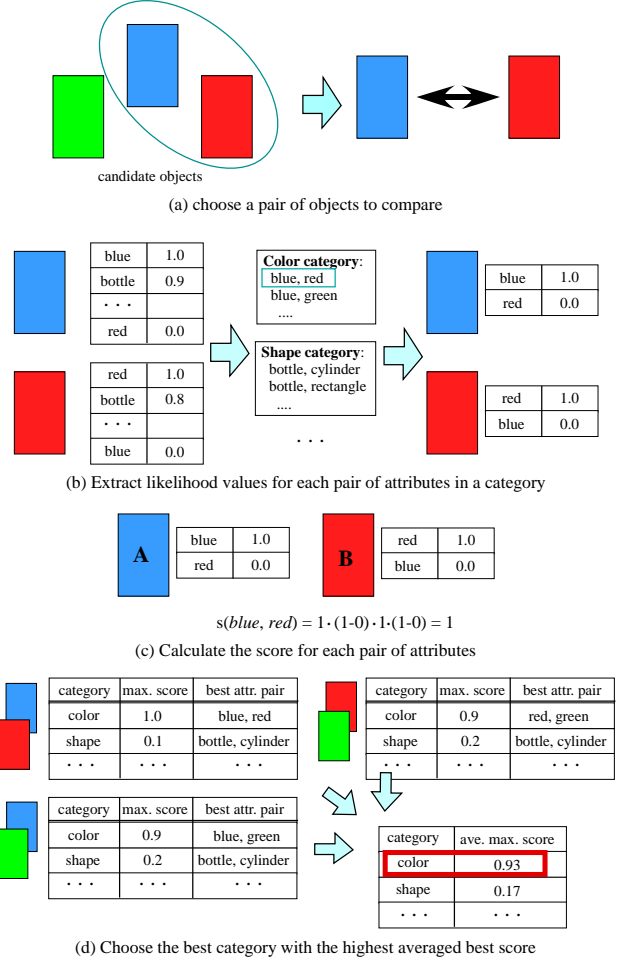


Fig. 2. Algorithm for category selection.

IV. IMPLEMENTATION AND EXPERIMENTS

A. Robot and software modules

We use Toyota’s HSR (Human Support Robot) [17] as a platform. The robot runs under ROS and in addition to HSR’s embedded modules, we implemented the following modules.

1) *Speech recognition*: We use PocketSphinx [18] speech recognition system for recognizing user’s commands. We defined a set of words to be used in this human-robot interaction. We also defined a grammar for three types of speech: Call, Command, and Answer as shown in Table III. Symbols in square brackets are options and those in angle brackets are variables indicating pre-defined words (e.g., attributes are pre-defined in Table I). We use Stanford CoreNLP [19] for morphological analysis.

2) *Object recognition*: We use YOLOv3 [14] for on-line object recognition as mentioned above. When only object type (cup, drink, and so on) is given in the command, the robot detects all objects of that type. When some attributes are included in the command (e.g., “bring me a paper cup.”), the robot additionally examines the detected objects with a specified attribute(s) and keeps the ones with that attribute(s) as candidates.

TABLE III
GRAMMAR DEFINITION.

Type	Description
Call	excuse me
Command	[could you please] bring [me] [<article>] [<attributes>] <target>
Answer	color is <color> state is <state> ... appearance is <appearance>



Fig. 3. Object images for the experiment.

B. Evaluation of generated queries for drinks

We currently do not have an appropriate dataset for evaluation specific to our bring-me task. We, therefore, compared the robot-generated and the user-generated queries for evaluating the quality of generated queries. Fig. 3 shows the set of objects used for this experiment. Among these objects, objects (a), (c), (d), and (e) are the ones used for training the network (see Sec. III-B), and the others (objects (b) and (f)) are new ones. Note that the user knows the complete set of attributes and uses them in generating queries.

We consider the following three cases: (1) two candidate objects, (2) three candidate objects, (3) three candidate objects with one attribute. We tested five scenarios for each case. Tables IV, V, and VI show the corresponding generated queries for respective scenarios.

A generated query is judged correct if it is included in the correct query list by the user. When a generated query has more than one attributes for a candidate, we judge it is correct if one of the attributes is included in the correct queries. The evaluation results are as follows.

Table IV shows the two object cases. All of the generated queries are correct since some effective attribute contrast always exists. Table V shows the three object cases. For the case of objects (a), (b), and (d) (in the third line of the table), the user judged that there are no effective attribute categories to discriminate the three objects at once, while the robot generates a query. The current algorithm always generates some query even if the evaluations of multiple categories are mostly similar. In such cases, generating a sequence of queries for multi-step discrimination will be necessary.

Table VI shows the result for three objects with one attribute cases. In such a case, if the robot judges that there is only one candidate which satisfies the given attribute, it does not generate any queries. For the cases with correct results, the attribute recognition for the candidates is correct, and the robot took a correct action. For the wrong cases, the recognition was not correct, especially for the objects not included in the training set, and the robot responded wrongly. We think these recognition failures might be solved by increasing the variety of training data.

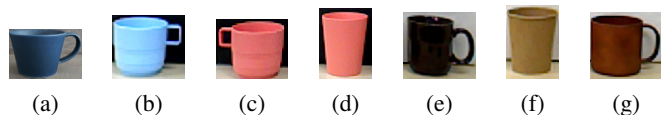


Fig. 4. Images of cups used for the experiment.

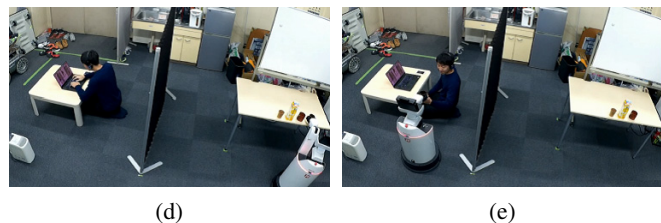
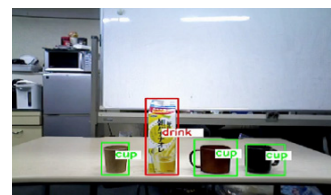
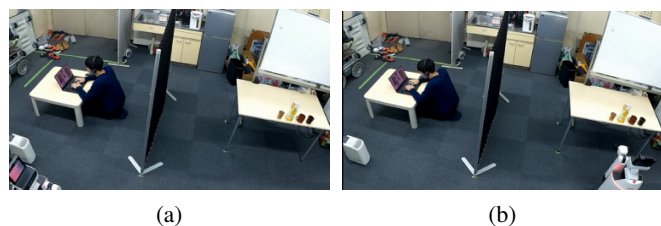


Fig. 5. Snapshots of the robot executing a bring-me task: (a) The user asks the robot to bring him a cup (more precisely, his cup); (b) The robot moves to the table to find candidates; (c) The robot recognizes four objects including three cups; (d) The robot holds the correct cup; (e) The robot hands over the cup to the user.

C. Evaluation of generated queries for cups

Recognition of drinks is not very difficult when they have distinctive labels on their surfaces. If it is not the case, however, we need to recognize objects based only on *natural* attributes. We thus tested our system with cups. Fig. 4 shows some of the cups. Table VII shows the results for three scenarios. Since we trained the network using these objects, the attribute extraction works well, which makes the query generation correct.

D. Robotic experiments

We set a scenario for the robotic experiments as follows:

- The robot has a map of the environment and all candidate objects are on a specific table.

TABLE IV
TWO CANDIDATE OBJECTS CASE.

objects	correct query (queries)	generated query	evaluation
(a), (c)	What is the color? one is red, one is green. What is the material? one is metal, one is plastic.	What is the material? one is metal, one is plastic.	correct
(a), (f)	What is the color? one is red, one is yellow. What is the material? one is metal, one is paper. What is the shape? one is cylinder, one is rectangle.	What is the color? one is red, one is yellow.	correct
(c), (d)	What is the shape? one is bottle, one is rectangle. What is the material? one is plastic, one is paper. How is the size? one is tall, one is small.	How is the size? one is tall, one is small.	correct
(b), (e)	What is the color? one is red, one is yellow. What is the material? one is paper, one is metal.	What is the color? one is red, one is yellow.	correct
(e), (f)	What is the color? one is blue, one is black. What is the color? one is blue, one is white. What is the material? one is metal, one is paper. What is the shape? one is cylinder, one is rectangle.	What is the material? one is metal, one is paper.	correct

TABLE V
THREE CANDIDATE OBJECTS CASE.

objects	correct query (queries)	generated query	evaluation
(a), (d), (f)	What is the color? one is red, one is green, one is yellow.	What is the color? one is red, one is green, one is yellow.	correct
(a), (e), (f)	What is the color? one is red, one is blue, one is yellow. What is the color? one is red, one is blue, one is white.	What is the color? one is red, one is yellow and white.	correct
(a), (b), (d)	<i>A single attribute category does not exist that can discriminate the three objects.</i>	What is attached? one is lid and label, one is lid and label, one is lettering.	inappropriate
(b), (d), (f)	What is the color? one is red, one is green, one is yellow.	What is the color? one is red, one is green, one is yellow.	correct
(c), (d), (e)	What is the material? one is plastic, one is paper, one is metal	What is the material? one is plastic, one is paper, one is metal.	correct

TABLE VI
THREE CANDIDATE OBJECTS AND ONE ATTRIBUTE CASE.

objects and attribute	predicted candidate(s)	correct query (queries)	actual candidate(s)	generated query	evaluation
(a), (d), (f) & 'paper'	(d), (f)	What is the color? one is green, one is yellow.	(d)	<i>no queries generated</i>	wrong
(a), (e), (f) & 'red'	(a)	<i>no queries generated</i>	(a)	<i>no queries generated</i>	correct
(a), (b), (c) & 'plastic'	(c)	<i>no queries generated</i>	(c)	<i>no queries generated</i>	correct
(b), (b), (f) & 'paper'	(b), (d)	What is the color? one is red, one is green. How is the size? one is tall, one is small.	(d)	<i>no queries generated</i>	wrong
(c), (d), (e) & 'green'	(c), (d)	What is the shape? one is rectangle, one is bottle. What is the material? one is paper, one is plastic. How is the size? one is tall, one is small.	(c), (d)	How is the size? one is tall, one is small.	correct

TABLE VII
CASES FOR CUPS.

objects	correct query (queries)	generated query	evaluation
(a), (b)	What is the material? one is ceramic, one is plastic.	What is the material? one is ceramic, one is plastic.	correct
(c), (d)	What is attached? one is grip, one is no-grip. How is the size? one is short, one is tall.	What is attached? one is grip, one is no-grip.	correct
(e), (f), (g)	What is the material? one is ceramic, one is paper, one is wood.	What is the material? one is ceramic, one is paper, one is wood.	correct

- Candidate objects are all drinks or cups.
- The user knows a complete set of objects (but does not

know which of them are on the table) and asks the robot to bring one specific object in his/her mind without

TABLE VIII

CATEGORY EVALUATIONS FOR THE SITUATION OF FIG. 5(C). THE MATERIALS ARE PAPER, WOOD, AND CERAMIC FROM LEFT TO RIGHT.

(a) Evaluation between the left and the center.		
category	best attribute pair	score
attachment	no-grip, grip	0.9995
material	paper, wood	0.7909
color	blue, blue	0.001292
size	tall, small	1.533e-8

(b) Evaluation between the left and the right.		
category	best attribute pair	score
attachment	no-grip, grip	0.9995
material	paper, wood	0.9977
color	blue, red	0.0006129
size	tall, small	1.533e-8

(c) Evaluation between the center and the right.		
category	best attribute pair	score
material	wood, ceramic	0.7602
color	blue, red	0.0006107
appearance	multi-colored, multi-colored	2.637e-7
attachment	no-grip, no-lid	7.122e-8

telling the attribute details, that is, some ambiguity in object identity always exists in the command.

Fig. 5 shows snapshots of a trial in the experiment. The user first gave a command “bring me a cup” to the robot (see Fig. 5(a)). The robot then went to the position in front of the table (its location is known) to find the cup (see Fig. 5(b)), and recognized three cups on the table (see Fig. 5(c)), noticing the command was ambiguous. It then generated a query “What is the material? one is paper, one is wood, one is ceramic.” and got an answer “Material is ceramic” from the user. Then the robot picked up the ceramic cup and brought it to the user to hand over (see Fig. 5(d)(e)). Table VIII shows the best three category scores for the three pairs of cups shown in Fig. 5(c). The table clearly shows that “material” is the best category for generating a query for resolving the ambiguity in the command in this situation.

V. CONCLUSIONS AND DISCUSSION

We have developed a query generation method for resolving ambiguities in the user’s command to a service robot. We deal with bring-me tasks and the method chooses the best category to ask based on the attribute contrast and the attribute categorization. The comparisons of robot-generated and human-generated queries and a robotic implementation show the effectiveness of the method.

The current dataset is limited in terms of object classes and object appearances in a class. Adding more data of various everyday objects for testing the system in a more realistic scenario is future work. The refinement of categories and attributes will also be necessary together. The type of ambiguities we currently consider is also limited to the one about the target object identity among possible candidates. To extend the system so that it can deal with a more variety of ambiguities due to speech recognition, object recognition, and location recognition is another future work.

ACKNOWLEDGMENT

This work is in part supported by JSPS KAKENHI Grant Number 17H01799. The authors would like to thank Toyota Motors Co. for providing an HSR for the experiments. They would also like to thank the members of Active Intelligent Systems Laboratory for the software modules used in the experiments.

REFERENCES

- [1] K.-T. Song, S.-Y. Jiang, and M.-H. Lin, “Interactive teleoperation of a mobile manipulator using a shared-control approach,” *IEEE Trans. on Human-Machine Systems*, vol. 46, pp. 834–845, 2016.
- [2] G. Watson, Y. Papelis, and K. Hicks, “Simulation-based environment for the eye-tracking control of tele-operated mobile robots,” in *Proceedings of Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016 and Space Simulation for Planetary Space Exploration 2016*, 2016, pp. 4:1–7.
- [3] Y. Ushiku, T. Harada, and Y. Kuniyoshi, “Automatic sentence generation from images,” in *Proceedings of 19th ACM Int. Conf. on Multimedia*, 2011, pp. 1533–1537.
- [4] K. Shih, S. Singh, and D. Hoiem, “Where to look: Focus regions for visual question answering,” in *Proceedings 2016 IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [5] Y. Li, C. Huang, X. Tang, and C. Loy, “Learning to disambiguate by asking discriminative questions,” in *Proceedings of 2017 Int. Conf. on Computer Vision*, 2017.
- [6] Y. Makihara, M. Takizawa, Y. Shirai, J. Miura, and N. Shimada, “Object recognition supported by user interaction for service robots,” in *Proceedings of the 16th Int. Conf. on Pattern Recognition*, 2002, pp. 561–564.
- [7] T. Spexard, S. Li, B. Wrede, M. Hanheide, E. Top, and H. Hüttenrauch, “Interaction awareness for joint environment exploration,” in *Proceedings of 16th IEEE Int. Symp. on Robot and Human Interactive Communication*, 2007.
- [8] R. Deits, S. Tellex, P. Thaker, D. Simeonov, T. Kollar, and N. Roy, “Clarifying commands with information-theoretic human-robot dialog,” *J. of Human-Robot Interaction*, vol. 1, no. 1, pp. 78–95, 2012.
- [9] R. Kiros, R. Zemel, and R. Salakhutdinov, “Multimodal neural language models,” in *Proceedings of 31st Int. Conf. on Machine Learning*, 2014.
- [10] J. Johnson, A. Karpathy, and F. Li, “Densecap: Fully convolutional localization networks for dense captioning,” in *Proceedings 2016 IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [11] S. Antol, A. Agrawal, J. Liu, M. Mitchell, C. Zitnick, D. Batra, and D. Parikh, “VQA: Visual Question Answering,” in *Proceedings of 2015 Int. Conf. on Computer Vision*, 2015.
- [12] K. Saito, A. Shin, Y. Ushiku, and T. Harada, “Dualnet: Domain-invariant network for visual question answering,” in *Proceedings of 18th IEEE Int. Conf. on Multimedia and Expo*, 2017.
- [13] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende, “Generating natural questions about an image,” in *Proceedings of 54th Annual Meeting of the Assoc. for Computational Linguistics*, 2016, pp. 1802–1813.
- [14] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” arXiv:1612.08242, 2016.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of 2016 IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [16] O. R. et al., “Imagenet large scale visual recognition challenge,” *Int. J. of Computer Vision*, vol. 115, pp. 211–252, 2015.
- [17] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, “Development of human support robot as the research platform of a domestic mobile manipulator,” *ROBOMECH journal*, 2019.
- [18] D. Huggins=Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, “Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices,” in *Proceedings of 2006 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2006.
- [19] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit,” in *Proceedings of 52nd Annual Conf. of the Association for Computational Linguistics: System Demonstrations*, 2014.