Ph.D. DISSERTATION

SEMANTIC SCENE UNDERSTANDING BASED HUMAN-ROBOT COOPERATION

의미론적 환경 이해 기반 인간 로봇 협업

BY

MOON JIYOUN

FEBRUARY 2020

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

Human-robot cooperation is unavoidable in various applications ranging from manufacturing to field robotics owing to the advantages of adaptability and high flexibility. Especially, complex task planning in large, unconstructed, and uncertain environments can employ the complementary capabilities of human and diverse robots. For a team to be effectives, knowledge regarding team goals and current situation needs to be effectively shared as they affect decision making. In this respect, semantic scene understanding in natural language is one of the most fundamental components for information sharing between humans and heterogeneous robots, as robots can perceive the surrounding environment in a form that both humans and other robots can understand. Moreover, natural-language-based scene understanding can reduce network congestion and improve the reliability of acquired data. Especially, in field robotics, transmission of raw sensor data increases network bandwidth and decreases quality of service. We can resolve this problem by transmitting information in the form of natural language that has encoded semantic representations of environments. In this dissertation, I introduce a human and heterogeneous robot cooperation scheme based on semantic scene understanding. I generate sentences and scene graphs, which is a natural language grounded graph over the detected objects and their relationships, with the graph map generated using a robot mapping algorithm. Subsequently, a framework that can utilize the results for cooperative mission planning of humans and robots is proposed. Experiments were performed to verify the effectiveness of the proposed methods.

This dissertation comprises two parts: graph-based scene understanding and scene understanding based on the cooperation between human and heterogeneous robots. For the former, I introduce a novel natural language processing method using a semantic graph map. Although semantic graph maps have been widely applied to study the perceptual aspects of the environment, such maps do not find extensive application in

natural language processing tasks. Several studies have been conducted on the understanding of workspace images in the field of computer vision; in these studies, the sentences were automatically generated, and therefore, multiple scenes have not yet been utilized for sentence generation. A graph-based convolutional neural network, which comprises spectral graph convolution and graph coarsening, and a recurrent neural network are employed to generate sentences attention over graphs. The proposed method outperforms the conventional methods on a publicly available dataset for single scenes and can be utilized for sequential scenes.

Recently, deep learning has demonstrated impressive developments in scene understanding using natural language. However, it has not been extensively applied to high-level processes such as causal reasoning, analogical reasoning, or planning. The symbolic approach that calculates the sequence of appropriate actions by combining the available skills of agents outperforms in reasoning and planning; however, it does not entirely consider semantic knowledge acquisition for human-robot information sharing. An architecture that combines deep learning techniques and symbolic planner for human and heterogeneous robots to achieve a shared goal based on semantic scene understanding is proposed for scene understanding based on human-robot cooperation. In this study, graph-based perception is used for scene understanding. A planning domain definition language (PDDL) planner and JENA-TDB are utilized for mission planning and data acquisition storage, respectively. The effectiveness of the proposed method is verified in two situations: a mission failure, in which the dynamic environment changes, and object detection in a large and unseen environment.

keywords: cognitive robotics, semantic scene understanding, 3D scene graph, mission planning, human-robot cooperation, natural language process

student number: 2014-22559

Contents

List of Tables

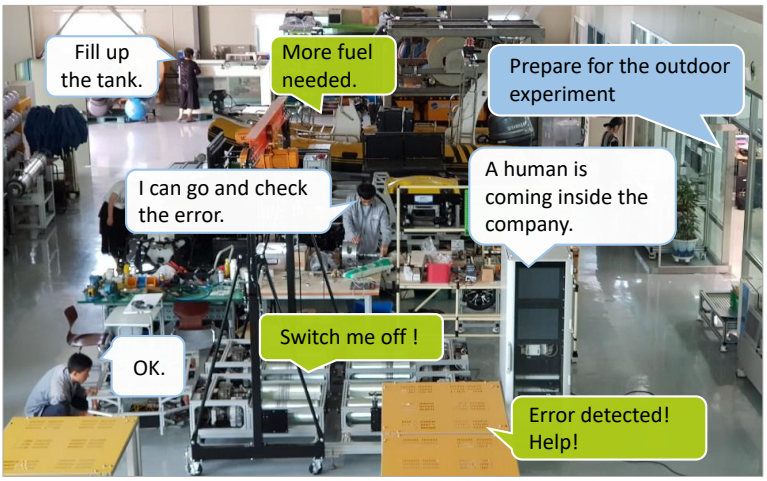
List of Figures

Chapter 1

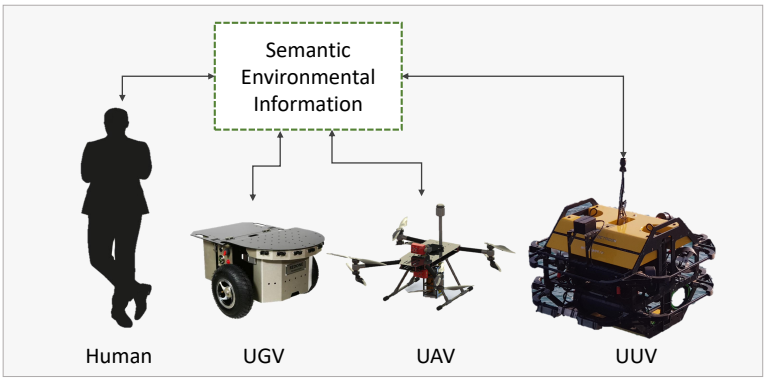
Introduction

1.1 Background and Motivation

A traditional robotic system can perform simple and repetitive tasks in well-structured environments. However, the application of robotic systems to various fields such as medicine, manufacturing, and exploration has led to an increasing demand of highly flexible robots that can work efficiently in an uncertain environment, which in turn has led to a considerable amount of attention being paid to such robots [1]. Combining the capabilities of humans such as adaptability, creativity, and intelligence and the abilities of robots such as rigidity, endurance, and speed can dramatically increase work efficiency [2]. Especially, cooperation between humans and heterogeneous robots can play a critical role in making robots adapt to an unstructured and dynamic environment [3]. Numerous algorithms have been developed to resolve issues such as sensing, perception to planning, control, and safety in human-robot cooperation. Among the various elements that need to be considered for a human-robot system, the most important is scene understanding based on natural language. This function of such a system can enable humans and robots to share information in a form that they both can understand, which is the most basic ability required for cooperation, as illustrated in Figure 1.1. Moreover, we can reduce network congestion by transmitting information in the form



(a)



(b)

(c)

Figure 1.1: *Cooperation between humans and heterogeneous robots: (a) natural language communication among heterogeneous agents (b) sharing semantic environmental information among heterogeneous agents (c) field robotics*

of natural language, which is a compact representation of environment information, instead of transmitting raw sensor data, which increase network bandwidth and decrease quality of service (QoS) when working in a large environment. Consequently, an improved QoS reliability can be guaranteed.

To perceive the surroundings of an environment in natural language, a scene graph, which is composed of detected objects (represented as nodes) and their relationships (represented as edges), language description and natural questions can be utilized. Figure 1.2 shows examples where each method is applied to human-robot cooperation.

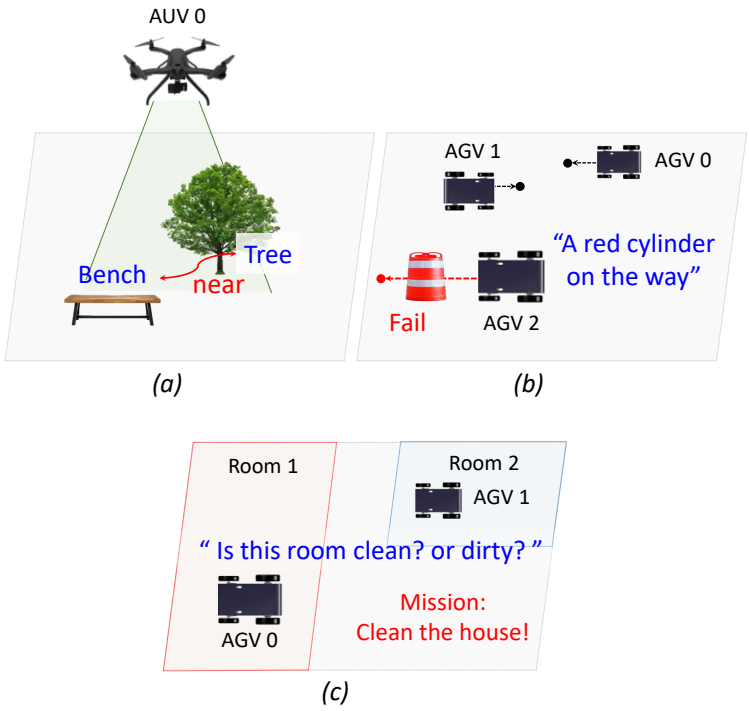


Figure 1.2: *Three examples of natural-language-based scene understanding in human-robot cooperation: (a) natural-language-based scene graph generation (b) language description (c) question generation*

- Scene graph generation can contribute toward information gathering in unseen and dynamic environments in a compact and communicable form. For example,

assume that a robot is located at a place where a human cannot approach. The generated scene graph can be used for humans to identify the location of the robot.

- Language description can help in recovering from mission failure, and this is particularly important because robot failures are inevitable. For example, assume that a robot has to travel to a certain position and wait for a human to load a package on it. However, a car may block the path of the robot, and therefore, the robot is now unable to approach the human. This leads to the robot failing the mission. In this case, the robot can inform the humans about the failure by describing the current situation and move to another location to complete the mission.

- Question generation can contribute toward the efficient operation of robots. For example, assume that a robot has to clean a house. Generally, humans command the robot regarding the rooms that need to be cleaned. However, if the robot can question itself regarding the tasks that it needs to perform or the status of the house and answer these questions by itself, human intervention will not be required to list all the tasks that the robot has to perform.

This dissertation introduces a human and heterogeneous robot cooperation scheme that is novel and includes sharing information regarding the surrounding environment in an interpretable form for both humans and robots. I generate a scene graph based on natural language, language description, and natural questions to perceive environments. Subsequently, a mission planning framework that includes semantic scene understanding is proposed for a team comprising humans and robots.

1.2 Literature Review

1.2.1 Natural Language-Based Human-Robot Cooperation

Natural language-based human-robot cooperation can be divided into three main processes as shown in Figure 1.3: instruction understanding, execution plan generation, and knowledge world mapping. The objective of this dissertation is to connect the two flows of studies, which is execution plan generation and knowledge world mapping.

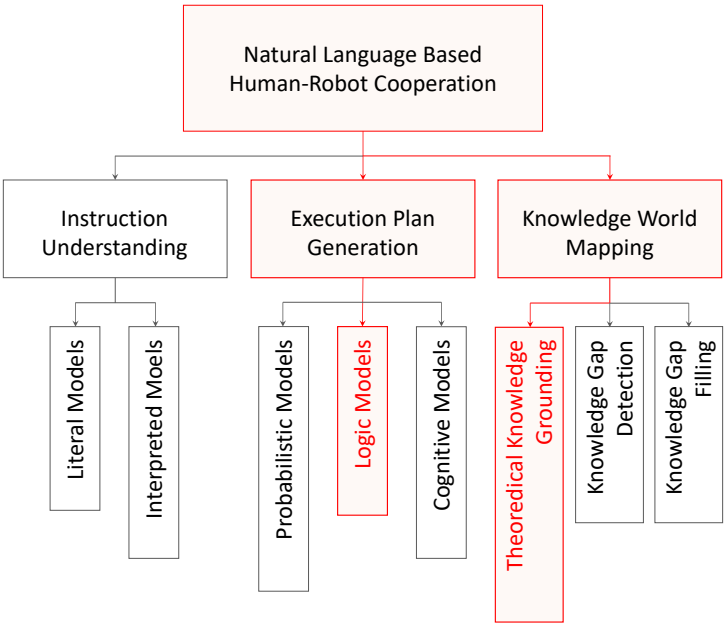


Figure 1.3: *Three main processes of natural-language-based human-robot cooperation: Instruction understanding, execution plan generation, and knowledge world mapping.*

1.2.2 Artificial Intelligence Planning

Artificial intelligence planning is a branch of artificial intelligence that aims to provide automation by generating a structure of actions that one or multiple agents use to

transition from an initial state to a desired goal state in a given environment. This is achieved by creating a model of the environment. The model aims to accurately represent the capabilities of the agent and the objects present in the environment, their attributes, as well as the relationships between them. In particular, the model includes an initial state, possible actions that affect the state, as well as the desired goal conditions.

A planner is used to identify one or more plans. A plan is a partially ordered set of actions which, once executed, are predicted by the model to achieve the goal condition. Typically, planners perform search through the state-space to find one or more action sequences that provide a transition from the initial state into a state in which the goal condition holds. These forward-search planners (e.g. [4]) are equipped with various heuristics to find solutions faster than having to explore every state in the state space, thus enabling their use for planning and re-planning online.

Automated Planning Framework

An automated planning has three important components: a state-transition system Σ, planner, and controller [5]. A state-transition system represents the real-world, in which agents perform automated planning. A function of Σ is defined as $S \times (A \cup E)$ where, states $S = \{s_0, s_1, s_2, \dots\}$, actions $A = \{a_1, a_2, a_3, \dots\}$, and events $E = \{e_0, e_1, e_2, \dots\}$. A planner generates plans, and a controller changes the state of systems by executing the actions. An automated planning framework is shown in Figure 1.4.

A planning language is an expression of the domain model encoding real-world environments in a logical form. Planning domain definition language (PDDL) has become the de facto standard language for domain independent mission planning [6]. It is a language developed from the STRIPS [7], which was mainly used for planning problems. It expresses the relationships and meanings of actions through preconditions and subsequent effects. Mission planning using PDDL is defined by two different files;

Table 1.1: An example of PDDL domain

```
(define (domain mobilerobot)
  (:types waypoint robot)
  (:requirements :strips :typing :fluents :disjunctive-preconditions :durative-actions)
  (:predicates  (robot_at ?v - robot ?wp - waypoint)  (connected ?from ?to - waypoint)
                (visited ?wp - waypoint)  (inspected ?wp - waypoint))
  (:functions  (distance ?wp1 ?wp2 - waypoint)  (battery ?v - robot))
  (:durative-action goto_waypoint
    :parameters (?v - robot ?from ?to - waypoint)
    :duration ( = ?duration 10)
    :condition ((at start (robot_at ?v ?from))
    :effect (and (at end (visited ?to)) (at start (not (robot_at ?v ?from)))
                (at end (robot_at ?v ?to))))
  (:action inspect
    :parameters (?v - robot ?wp - waypoint)
    :duration ( = ?duration 60)
    :condition (and (at start (>= (battery ?v) 1)) (at start (robot_at ?v ?wp))
                (at start (visited ?wp)))
    :effect (and (at end (inspected ?wp)) (at start (decrease (battery ?v) 1))))
  (:action charge
    :parameters (?v - robot ?wp - waypoint)
    :duration ( = ?duration 10)
    :condition (and (over all (robot_at ?v ?wp)) (over all (<= (battery ?v) 0)))
    :effect (at end (assign (battery ?v) 2)))))
```

Table 1.2: An example of PDDL problem

```
(define (problem task)
  (:domain mobilerobot)
  (:objects wp0 wp1 wp2 wp3 wp4 wp5 - waypoint
            kenny - robot)
  (:init  (robot_at kenny wp0)  (= (battery kenny) 2)  (visited wp0))
  (:goal  (and (inspected wp0) (inspected wp1) (inspected wp2)
               (inspected wp3) (inspected wp4) (inspected wp5)
               (= (battery kenny) 2))))
```

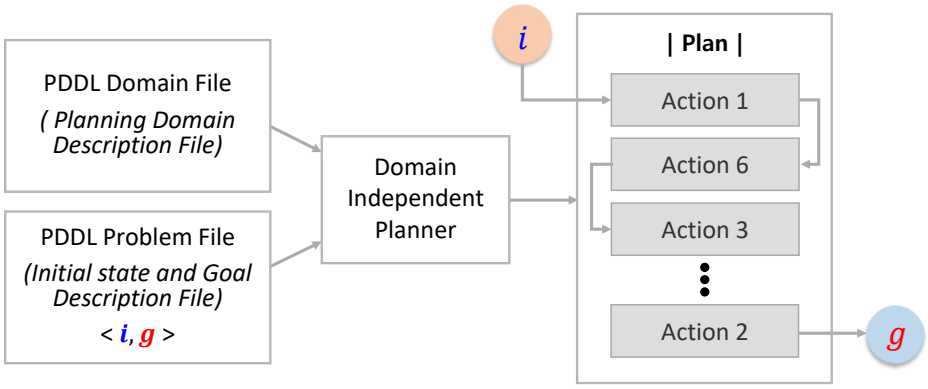


Figure 1.4: *Automated planning framework*

one is the domain file, and the other is the problem file. An example of each file is presented in Table 1.1 and Table 1.2. The domain file consists of object types, predicates, functions, and actions, and the problem file consists of objects, initial states, and goal states.

Offline and online planning

Planning is divided into two different types. One is offline that generates plans in advance and subsequently performs actions, and the other is online, which generates plans when the actions are being performed. Figure 1.5 shows the conceptual models of offline planning and online planning. Offline planning has a one-way connection between the planner and controller. Therefore, it cannot provide feedback about the current situations or observations. In other words, it only generates plans using initial state and objectives, while not considering current dynamic situations critical in real-world situations. Online planning has a two-way connection between the planner and controller and is therefore applicable in a dynamic environment. It can dynamically generate plans according to environmental changes by continuously updating sensor data. However, online planning using only raw sensor information has a limitation in that it needs to design various algorithms based on the situation. It is therefore

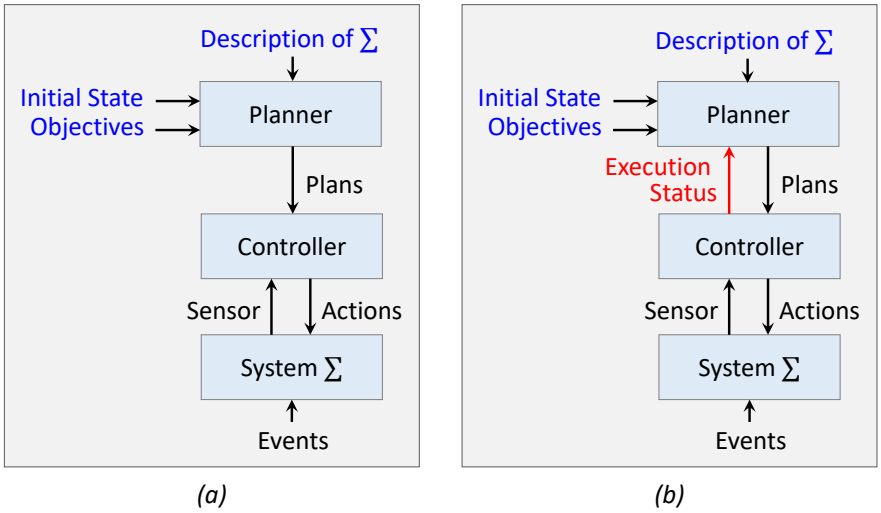


Figure 1.5: *Conceptual models of offline and online planning systems: (a) offline planning (b) online planning*

vulnerable to the type of situation. This dissertation introduces a mission planning method based on a semantic scene understanding that is robust to diverse situations.

Planning representation languages

The representation languages for automatic planning are detailed in Table 1.3 [6]. NDDL was developed by NASA for the EUROPA2 planning system. It became a practical approach for searching engines by replacing state and behavior with time constraints. HTN-PDDL uses hierarchical elements to perform hierarchical tasks. PDDL is the standard language used for domain-independent planners and has been adopted as the standard for domain modelling languages. The latest version of PDDL is PDDL3.1. It presents object-fluents, which can not only handle any object-type. In this dissertation, we utilize PDDL2.1, which is widely used in mission planning in robotics. It provides numerical fluents, plan-metrics, and durative/continuous actions.

Table 1.3: Summary of planning domain definition languages

PDDL	Standardized syntax for STRIPS
PDDL+	Models continuous time-dependent effects
PDDL2.1	Extension of PDDL to numeric fluents and temporal planning
PDDL3.0	Introduced hard and soft constraints for preference-based planning
PDDL3.1	Introduced functions and object fluents
PPDDL	Extension of PDDL2.1 for probabilistic planners
HTN-PDDL	Extension of PDDL for hierarchical task networks
STRIPS	Sublanguage of PDDL. Unknown literals are false (closed-world)
OCL	Object-Centered representation
ANML	Combines best aspects of PDDL and NDDL
RDDL	STRIPS + functional terms, leading to higher expressiveness
ADL	An extension of STRIPS to include negative conditions
NDDL	Constraints between those intervals as states and actions

1.3 The Problem Statement

Among the three models to generate an execution plan for cooperation between humans and heterogeneous robots, logic models, which is also known as symbolic planning, exhibited good performance in generating a complete sequence of actions by combining available skills that each agent can perform. However, if one action cannot be performed because of environmental changes, the remaining actions cannot be performed. It will lead to mission failure. This weak environment adaptation limits symbolic planning in practical applications. However, most studies on symbolic planning have focused on algorithms that determine an optimal plan, which maximizes the overall utility and minimizes costs, while hardly considering information sharing issues between humans and robots that can resolve the environment adaptation problem. To share information between humans and robots, the most important element that needs to be considered in advance is natural-language-based scene understanding, which is one study flow of knowledge world mapping for human-robot cooperation. Deep learning techniques are widely utilized in perceiving environments in natural

language. However, it has hardly been applied to symbolic planning algorithms.

1.4 Contributions

The narrative of this dissertation is presented through a series of published works, prefaced by a review of the current state of the field. The main contribution of this dissertation is the utilization of natural-language-based human-robot cooperation for scene understanding. Scene graph, language description, and natural questions are generated to understand the surrounding environment and are applied to mission planning for a team comprising humans and heterogeneous robots.

In this dissertation, first, I propose three different ways of perceiving the environment in natural language. Subsequently, a framework that integrates a symbolic planning algorithm and deep learning techniques is proposed. Experiments are performed to verify the effectiveness of the proposed methods for human-robot cooperation in a dynamic and large environment.

- **Chapter 2** describes the generation of a semantic graph by using a visual genome dataset and data collected using an actual mobile robot. A convolutional neural network and a region proposal network detect and perceive objects in the surrounding environment. Subsequently, a recurrent neural network is used for relationship inference. The entire process is verified through experiments.

- **Chapter 3** introduces a novel natural language processing method using a 3D semantic graph map. A graph convolutional neural network and a recurrent neural network are used to generate a description of the map. A natural language sentence focusing on objects over a 3D semantic graph map can be eventually generated. I validate the proposed method by using a publicly available dataset and compare it with conventional methods.

- **Chapter 4** introduces a method to generate natural questions using object-oriented

semantic graphs. First, a graph convolutional neural network with a graph coarsening algorithm extracts features from the graph. Subsequently, a long short-term memory generates natural questions from the extracted features. Using graphs, natural questions can be generated for both single and sequential scenes. The proposed method outperforms conventional methods on a publicly available dataset for single scenes and can generate questions for sequential scenes.

- **Chapter 5** describes a method that can utilize natural language in mission planning. The knowledge provided by humans in the form of natural language can fulfill the missing robot's workspace knowledge and help recover from mission failure. A natural language technology, which transforms underspecified sentences into their formal forms, and resource description framework (RDF) graph-based ontologies, are utilized. Experiments were conducted for two categories of scenarios for validation. One deals with a partially known workspace, and the other with the acquisition of missing knowledge.

- **Chapter 6** presents a cooperation framework that integrates deep learning techniques and symbolic planner for heterogeneous robots is proposed. Neural networks are employed for natural-language-based scene understanding to share environmental information among robots. A sequence of actions for each robot are generated by using the PDDL planner. JENA-TDB is used for data acquisition store. The proposed method is validated through a simulation performed with an unmanned aerial vehicle and three unmanned ground vehicles.

1.5 Dissertation Outline

The dissertation consists of seven chapters. Chapter 2 introduces a scene graph generation algorithm. Chapters 3 and 4 present natural language sentence generation methods using graph map. Chapter 5 presents a planning system using natural language for artificial intelligence to organize actions in order to achieve goals. Chapter 6 proposes

a human-robot cooperation framework that integrates natural-language-based scene understanding and PDDL planning. Chapter 7 summarizes the contributions of this dissertation and discusses future work.

Chapter 2

Natural Language-Based Scene Graph Generation

2.1 Introduction

Semantic scene understanding is important in human-robot cooperation. In addition to simply representing the surrounding environments with lines and planes, recently, research on extracting meaningful information such as topography and objects has been widely conducted [8]. [9] proposed an RGB-D simultaneous localization and mapping (SLAM) framework with object-level entities. [10] designed a monocular SLAM framework for robust object-oriented map generation in a dynamic environment. However, most research on semantic scene understanding has focused only on improving mapping performance, instead of utilizing a generated semantic map. [11] efficiently learned human-centric models with a framework composed of grounded natural language descriptions and semantic maps. [12] improved robot navigation performance by demonstrating robotic navigation algorithms based on semantic maps and natural language interfaces. Since these studies only use the generated semantic map with detected objects, they have a limitation – the information about relationships between objects which are not included in the map is hardly utilized.

To ensure cooperation through seamless human-robot communication, it is necessary to represent detected objects and their relationships in the form of natural lan-

guage. In the field of computer vision, various forms of natural language scene understanding, such as image captioning [13], visual question answering [14], and scene graph generation [15] have been conducted. There have been many studies on image captioning and visual question answering, but only a few on scene graph generation, which can extract semantic meanings of the relationships between objects. [16] generated image captions and a scene graph by constructing a multi-level scene description network. [17] proposed generative adversarial networks that can infer the relationships between objects as well as their properties. These methods only utilize publicly available datasets when generating scene graphs.

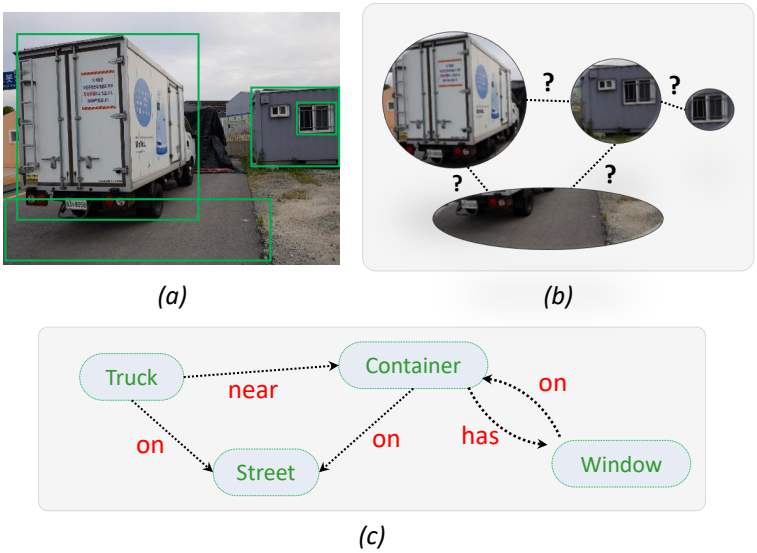


Figure 2.1: *Scene graph generation: (a) Object detection with an image (b) A graph map obtained with semantic SLAM (c) Natural-language-based scene graph*

A robot detects objects from surrounding environments with images obtained from sensors and object detection algorithms as shown in Figure 2.1 (a). Then, a semantic graph map is generated through semantic SLAM algorithms as illustrated in Figure 2.1 (b). Nodes of the graph map consist of feature and position information, while edges are not defined. Thus, it is difficult to infer the semantic meaning of the relationship

between objects using the semantic graph map. In this study, I generate scene graphs, which can predict not only objects but also the relationship between them using natural language as shown in Figure 2.1 (c). Generating a scene graph is composed of three major processes: object detection, perception, and relationship inference. I detect and perceive objects using a convolutional neural network (CNN) and a region proposal network (RPN). I then perform graph inference using a recurrent neural network (RNN). The process of scene graph generation is verified through experiments using a publicly available dataset and data that is obtained using mobile robots.

2.2 Related Work

This dissertation is part of three kinds of research: human-robot cooperation, semantic scene understanding, and relationship finding.

Human-robot cooperation

Human-robot cooperation is a highly researched subject. [18] presented an object acquisition algorithm by which humans and robot can cooperate. [19] proposed a human-robot collaborative disassembly system using deep reinforcement learning, incremental learning, and transfer learning techniques. [20] constructed a cyber-physical system so that humans and robots can work together safely. However, since these studies are designed for specific situations, an unexpected situation is difficult to handle. Communication between humans and robots using natural languages can be used to overcome such situations. [2] proposed a generalized grounding graph framework that allows a robot to ask a human for help when it fails a mission. [21] increased the efficiency of navigation through conversation between people and robots. However, these studies only consider natural language related cooperation methods, assuming that an understanding of the surrounding environment is sufficiently achieved. In this dissertation, a method of natural-language-based scene understanding is introduced.

Semantic scene understanding

The simplest method of semantic scene understanding is to list all the objects in an image. [22] introduced the you only look once (YOLO) detection system, which can detect multiple objects in real time. [23] improved the single shot multibox detector to enhance the performance of multi-object detection in complex traffic environments. However, for human-robot cooperation, the environment should be represented in natural language so that it can be understood by humans. [24] proposed dense a semantic embedding network to generate a natural language sentence containing the latent concept of an image. [25] generated a sentence focusing on the attributes of the objects in the image. [26] presented a bottom-up and top-down attention mechanism for object-oriented sentences. Since most image description methods ambiguously generate a sentence about the whole image, it is difficult to infer descriptive indications of specific image regions. Moreover, most of the generated sentences are related to detected objects. Here, I find specific regions and infer natural language descriptions of both objects and their relationships.

Relationship finding

Recently, there has been increased interest in finding the relationship between objects [27, 28]. Rather than simply detecting the objects in the image, researchers try to find relationships between them in triple forms, such as <subject-predicate-object>. [29] proposed a visual phrase detector to find relationships between objects. However, since these existing methods are based on the classification method, only limited relationships can be found. On the other hand, [30] used a language prior and [31] used a deep relational network to find various relationships between objects. [32] constructed a visual phrase guided convolutional neural network to simultaneously solve three problems – image captioning, object detection, and visual relationship detection. Most of these experimental methods utilize publicly available datasets. In this study, I generate a scene graph using both publicly available datasets and data obtained using a mobile

robot.

2.3 Scene Graph Generation

This section describes an image-based scene graph generation method. A scene graph is composed of detected objects represented as nodes and the relationship between objects as edges in natural language. Relationship finding is a critical component of semantic scene understanding. Two images featuring the same objects can differ greatly depending on the relationship between the objects. For example, in Figure 2.2, the two images, both consisting of cups and flowers, have different descriptions based on the location of the flowers. The overall process is composed of graph construction and graph inference as illustrated in Figure 2.3.

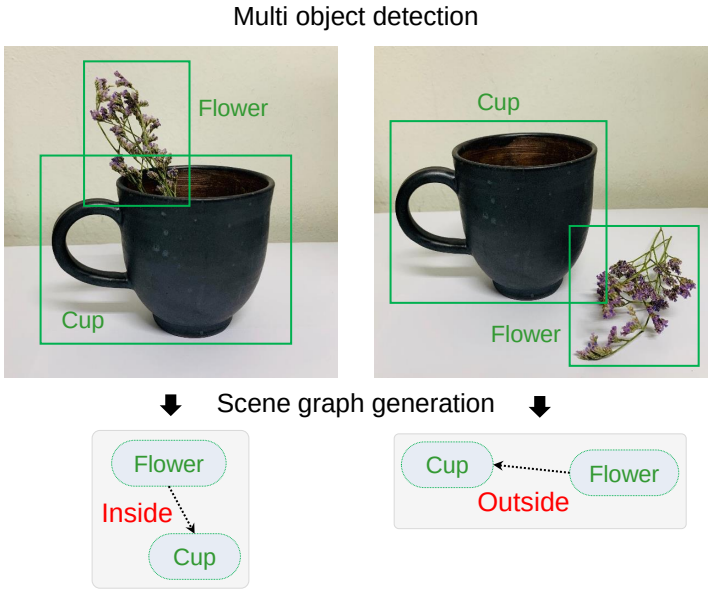


Figure 2.2: *The importance of finding object relationships: both pictures are composed of the same objects, but the situation is significantly different depending on the relationship between them.*

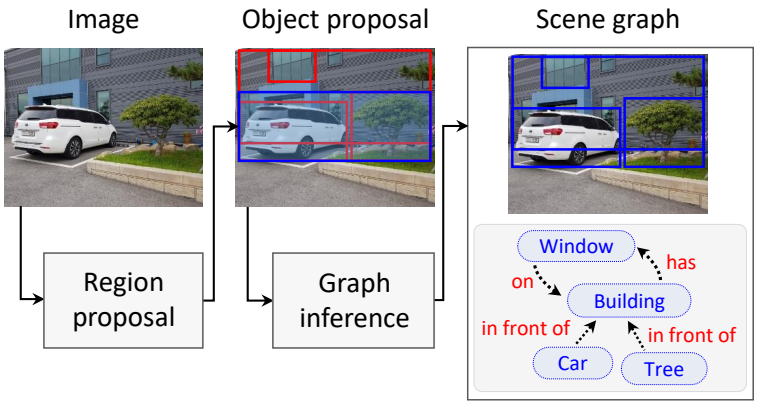


Figure 2.3: *Process of scene graph generation: Scene graph generation is composed of region proposal and graph inference.*

2.3.1 Graph Construction

In the visually grounded scene graph, object bounding boxes are marked as nodes and the pair-wise relationship between objects as edges. Object bounding boxes are provided by a dataset or can be obtained with object detection algorithms. In this study, I obtained bounding box proposals from an image using the RPN [33] as shown in Figure 2.4. Then, I predicted object class labels, bounding box offsets, and relationship-centric words for every object pair.

The image with n proposal bounding boxes is represented as a graph $G = (V, E)$ composed of nodes $v_i \in V$ and edges $e_{ij} = (v_i, v_j) \in E_{ij}$. I denote all variables related to the graph as $g = (v_i^{class}, v_i^{bbox}, e_{ij} \mid i = 1 \dots n, j = 1 \dots n, i \neq j)$. Given C is a set of object classes and R is a set of relationship types, $v_i^{class} \in C, v_i^{bbox} \in \mathbb{R}^4, e_{ij} \in R$.

2.3.2 Graph Inference

I inference natural language words corresponding to each node and edge of the graph. The position and type of objects are predicted as nodes while the relationship between

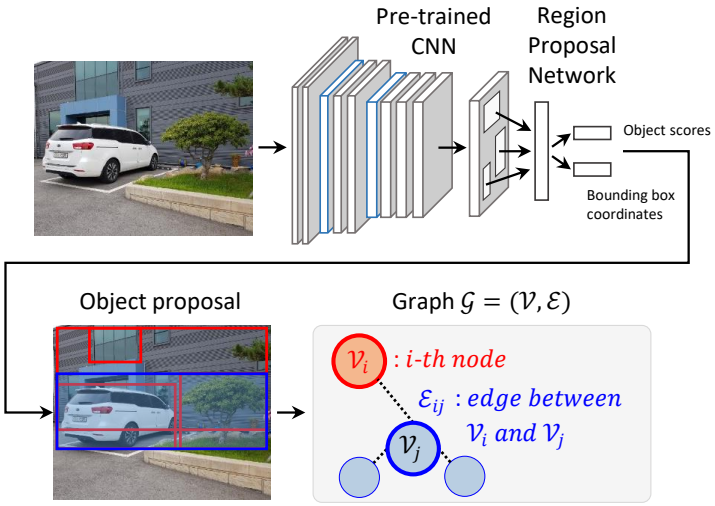


Figure 2.4: *Process of object detection: Pre-trained CNN and RPN are used for detecting objects. Then, the initial graph is constructed using the object proposal.*

nodes are predicted as edges. The process of obtaining the optimal scene graph G^*, which represents the surrounding environment precisely, is as follows.

$$g^* = argmax_g Pr(g \mid I, B_I) \tag{2.1}$$

$$Pr(g \mid I, B_I) = \prod_{i \in V} \prod_{j \neq i} Pr(v_i^{class}, v_i^{bbox}, e_{ij} \mid I, B_I) \tag{2.2}$$

where, I and B_I are input images and bounding box proposals. In my study, I closely follow an iterative message passing model [34] based on gated recurrent units (GRU), one of the generic RNN modules, for inference approximation. The network architecture is shown in Figure 2.5. The features of a bounding box in the image are used as initial input of the node GRU, and the union region features between the two bounding boxes are applied as the initial input of the edge GRU. The output of the GRU is fed into the next message pooling iteration. This process is repeated to generate a scene graph.

The hidden state of the GRU indicates the current state of each node and edge. The same update rule is applied to all nodes and edges. The hidden state of node v_i is

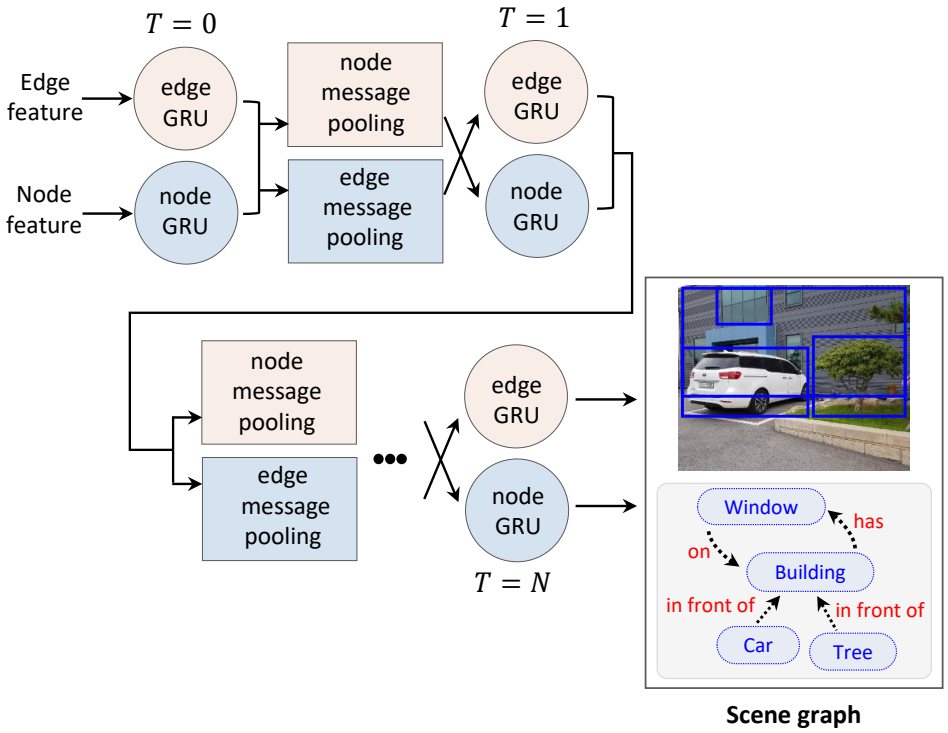


Figure 2.5: *Overall network architecture*

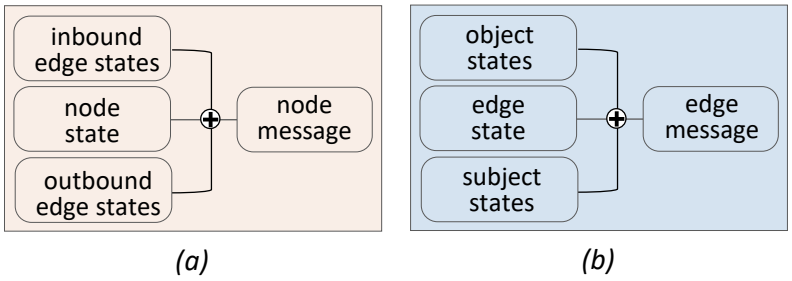


Figure 2.6: *Message pooling: (a) Node message pooling (b) Edge message pooling*

h_i and the hidden state of edge e_{ij} is h_{ij}. The inference approximation is formulated using the mean field distribution as follows:

$$Q(g \mid I, B_I) = \prod_{i=1}^{n} Q(v_i^{class}, v_i^{bbox} \mid h_i) Q(h_i \mid f_i^v) \tag{2.3}$$
$$\prod_{j \neq i} Q(e_{ij} \mid h_{ij}) Q(h_{ij} \mid f_{ij}^e)$$

where $Q(g \mid \cdot)$ is the probability of the graph variables g. I assume that each iteration is only affected by the current state. f_i^v and f_{ij}^e are features of node v_i and edge e_{ij}, respectively.

The graph characteristic of the bipartite structure, in which the neighbor of a node is an edge and the neighbor of an edge is a node, is utilized for graph inference. Message pooling is applied to two disjoint graphs, node-centric graphs and edge-centric graphs. Node message pooling uses the inbound and outbound edge states with a node as shown in Figure 2.6 (a). Edge message pooling uses the object states with an edge as shown in Figure 2.6 (b). The message pooling function computes the node message m_i and the edge message m_{ij}, which connects node i and node i as follows:

$$m_i = \sum_{j:i \to j} \sigma(W_1^T [h_i, h_{ij}]) h_{ij} + \sum_{j:j \to i} \sigma(w_2^T [h_i, h_{ji}]) h_{ji} \tag{2.4}$$

$$m_{ij} = \sigma(W_3^T [h_i, h_{ij}]) h_i + \sigma(w_4^T [h_j, h_{ij}]) h_j \tag{2.5}$$

where, $Sigma$ is the activation function, and w_1, w_2, w_3, w_4 are learnable parameters. I find relevant information between messages through message pooling. The output of the node message pooling is fed as input to the edge GRU, and the output of the edge message pooling is fed as input to the node GRU. This process is repeated to precisely predict the natural language words corresponding to the nodes and edges of the graph.

2.4 Experiments

I generated initial graphs using a region proposal algorithm and a scene graph using a graph inference method. Experiments were conducted using both a visual genome

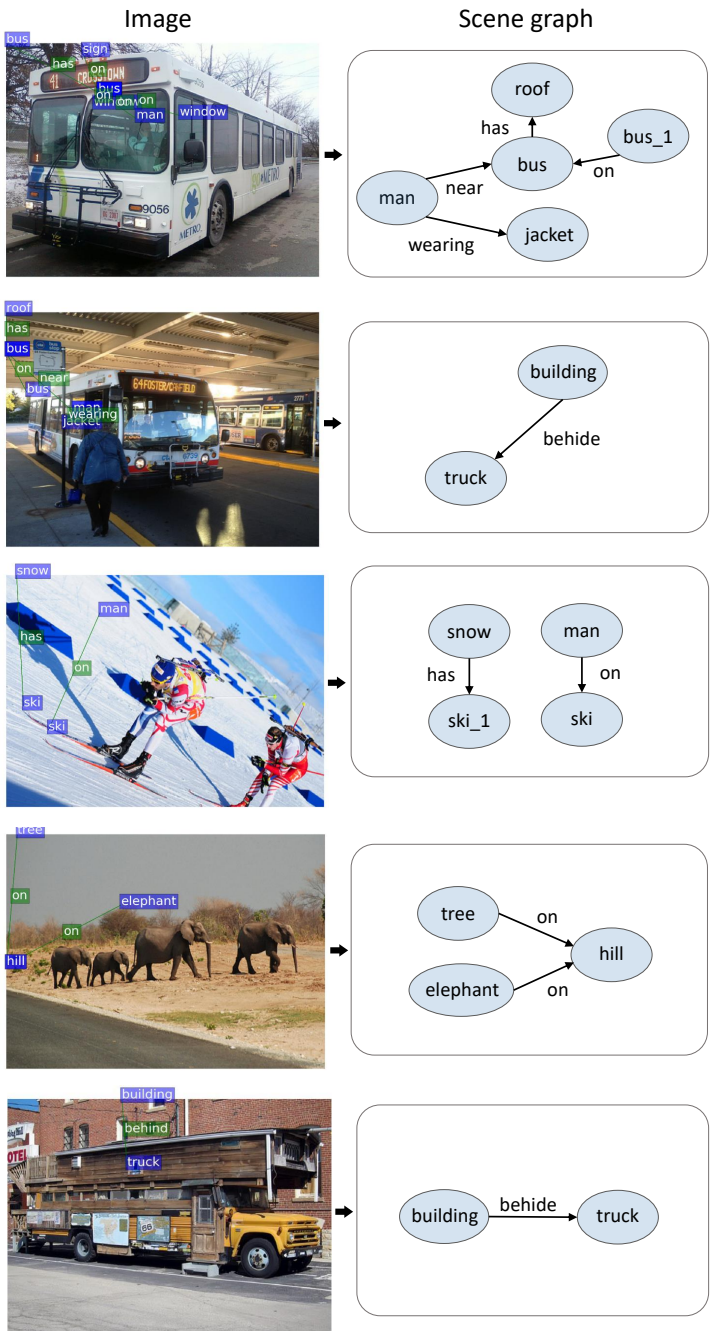


Figure 2.7: *Results of scene graph generation using the visual genome dataset*

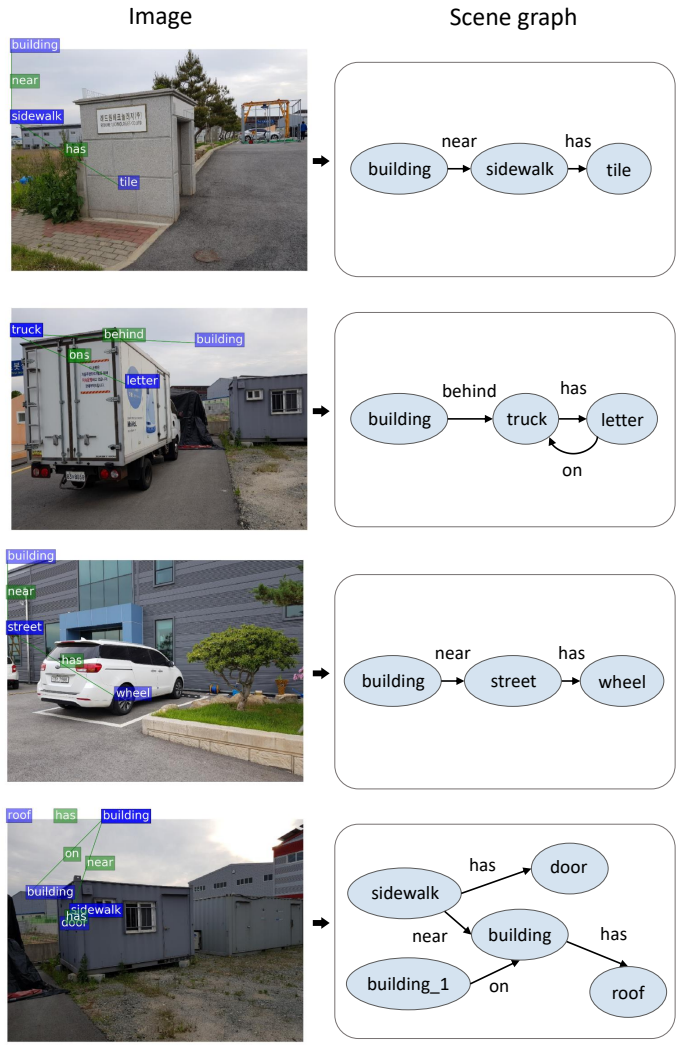


Figure 2.8: *Results of scene graph generation using images gathered by a robot*

dataset [35] and data obtained using a robot. Visual genome datasets consist of object labels and annotations of object relationships. The dataset has $1,531,448$ relationships for $108,077$ images; I utilized 70% of the data for training and 30% for testing. I used a four-wheel mobile robot to gather real world images for scene graph generation. An image was encoded as features using a pre-trained network, VGGNet 16,after which initial graphs were generated using a pre-trained RPN. The Tensorflow library was used for network construction, and the network was trained using a momentum optimizer with 0.9 momentum.

The results of scene graph generation are shown in Figure 2.7 and Figure 2.8. I successfully generated scene graphs for both the visual genome dataset and the images obtained using a mobile robot. All results are in triplets <subject-predicate-object>. Experimental results show that the relationship between objects is bidirectional; for example, <truck-has-letter> and <letter-on-truck>. However, because scene graphs are generated with images that rarely contain 3D information, there is the limitation of incorrect position information. For example, in Figure 2.8, <building-near-sidewalk> is generated even though the building and sidewalk are far apart.

2.5 Summary

In this study, I successfully generated natural-language-based scene graphs for human-robot cooperation. The CNN and RPN extract features from images, and generate the initial graph. Then, a GRU-based iterative message passing technique is used for graph inference. The method of generating visually grounded scene graphs using both a visual genome dataset and data obtained using a mobile robot was also verified. In the future, I plan to utilize a scene graph as a scene understanding method for human-robot communication in mission planning.

Chapter 3

Language Description with 3D Semantic Graph

3.1 Introduction

A natural language description for working environment understanding is an important component in human-robot communication. Although 3D semantic graph mappings are widely studied for perceptual aspects of the environment, these approaches hardly apply to the communication issues such as natural language descriptions for a semantic graph map. There are many researches on workspace understanding over images in the field of computer vision, which automatically generate sentences while they usually never utilize multiple scenes and 3D information. In this dissertation, I present a novel natural language description method using 3D semantic graph map.

3.2 Related Work

Working environment understanding using natural language is an important issue in human-robot cooperation. The efficiency of work can be improved by complementing human-robot understanding capabilities [2]. For example, a person may assign a mission to a robot to go into another room and bring a cup on a desk, but if the cup is on a chair, the robot will fail the mission. However, if the robot is able to describe the

surrounding environment in a natural language, not only the reason for the failure of the mission can be notified to the person, but also the success rate of the mission can be drastically increased by requesting a changed mission. However, there is a lack of consideration on human-robot cooperation compared to simultaneous localization and mapping (SLAM) and collision avoidance, which are focusing on improving robot motion capabilities [36]. Matuszek et al. [37] and Chen et al. [38] utilize natural language for communication between human and robot. However, their methods assume that the working environment is fully understood in prior [39]. Moreover, generating natural language descriptions for workspace understanding is less considered than generating natural language requests for giving commands. I here propose a natural language description method for working environment understanding using object-oriented semantic graph map composed of multiple scenes.

In robotics, the conventional working environment map generally uses features such as corners, surface patches, or lines, which hardly infer semantic contents [8]. As the object recognition technology [40] has been actively developed recently, semantic mapping algorithms using graphical model have been widely studied [8], [11]. These graph maps contain semantic information such as features and positions of objects meaningful to both human and robot. Galindo et al. [41] not only utilized objects in the map, but also proposed a spatial hierarchy concept which infers the room category considering the type of objects in the room. Hemachandra et al. [39] combined the low-level map data with natural language words to give more precise semantic information to the map. However, these methods are only focusing on the way of adding higher-level features such as object or room types on a map rather than generating natural language sentences to interact with human. Therefore, a natural language description method which can fully express the working environment needs to be studied.

Researches of understanding the environment through natural language have been developed significantly in the field of computer vision [42], [43]. Lin et al. [44] proposed a rule-based algorithm which parses a 3D scene using RGB-D images and gener-

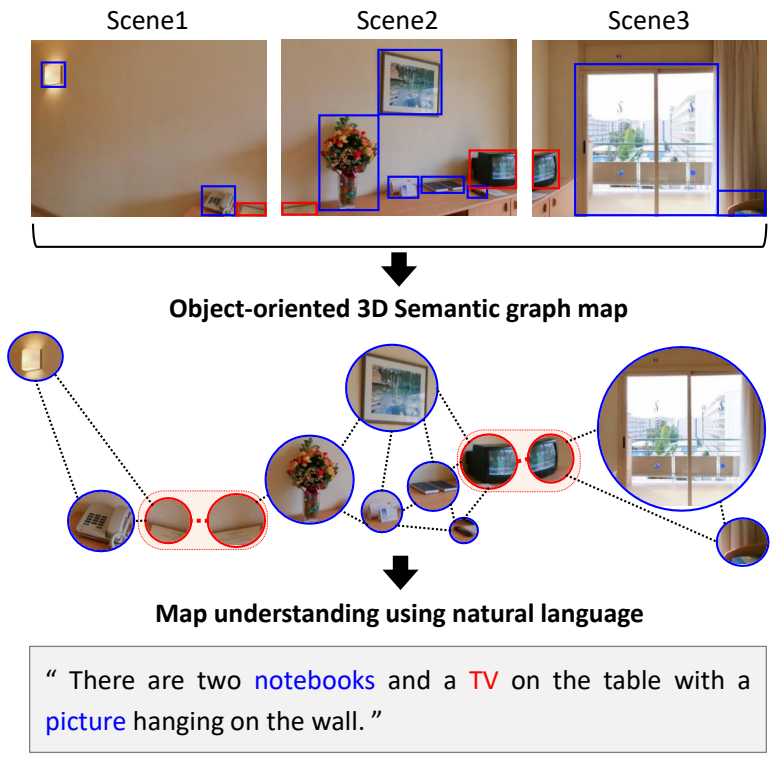


Figure 3.1: *An object-oriented 3D semantic map understanding using natural language description: I use multiple scenes to generate a 3D semantic object-oriented graph map. The nodes of this graph consist of object information in each image, and the graph is extended through overlapped objects as the scene increases. Then, a natural language sentence is generated describing multiple scenes focusing on objects using the map.*

ates corresponding sentences. Since this method is based on the hand-crafted method, it hardly generates various sentences on a single image. On the other hand, Vinyals et al. [45] and Karpathy et al. [46] generated various natural language sentences based on multimodal recurrent neural network and long short-term memory (LSTM), respectively. Since these methods are based on images, it is hard to consider 3D information and multiple scenes simultaneously.

3.3 Natural Language Description

The goal of this dissertation is to generate a sentence using object-oriented 3D semantic graph map composed of multiple scenes as shown in Figure 3.1. The works that are similar to my annotate natural language words to scene graphs generated by images [32], [47]. Li et al. [48] finds a word describing the entire scene using graph generated by the detected objects. Li et al. [16] proposed a multi-level scene description network that generates a graph and annotates the corresponding word using images. However, these works are focusing on annotating words to graphs, not generating sentences. Therefore, I generate a natural language sentence with focused over an object-oriented 3D semantic graph map.

The composition of this dissertation is as follows. Section 2 discusses the graph based 3D semantic map architecture using RGB-D images. Section 3 shows a natural language sentence generation of the 3D semantic graph map using graph convolutional neural network (GCN) and recurrent neural network (RNN). Section 4 presents the validation of the proposed method through publicly available datasets with state-out-the-art algorithms. Section 5 discusses conclusion and future work.

3.3.1 Preprocess

Conventional methods such as Vinyals et al. [45] and Karpathy et al. [46] can be used to generate a sentence describing multiple scenes using panorama images [49]. How-

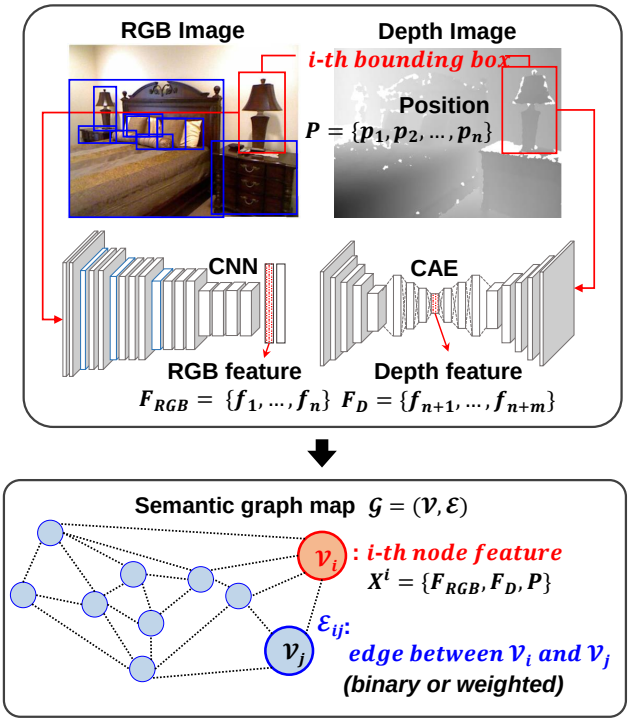


Figure 3.2: *An object-oriented semantic graph map generation based on RGB-D image: A semantic graph map $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed using features of detected objects in RGB-D images. A node feature X^i comprises features of RGB image F^i_{RGB}, features of depth image F^i_D, and position of the bounding box P^i. F^i_{RGB} is extracted by pre-trained CNN, and F^i_D is extracted by a CAE. An edge $\mathcal{E}_{ij}$ denotes the relation between two objects.*

ever, in order to apply the panorama image directly, the image should be reshaped into the smaller size appropriate for their proposed algorithms. Thus, features of objects in each scene could be partially lost. In addition, since these methods are processed with 2D data, it has a disadvantage that 3D data such as RGB-D image can be hardly applied. To utilize 3D data by extending existing methods, features of depth images can be extracted by convolutional auto-encoder (CAE) [50] as a pre-trained neural network [51] for RGB images. However, this approach also has a limitation that multiple images can be hardly applied to the network that is trained only with single images.

To overcome these limitations, I use an object-oriented 3D semantic graph map as shown in Figure 3.2. This graph map has features of detected objects in the scene as nodes and pair-wise relationships of each object as edges. Thus, 3D information of objects from multiple scenes can be efficiently stored as nodes increase. When all objects, in a scene, are represented in the form of nodes, edges can be connected using Delaunay triangulation algorithms based on the distance between objects or the similarity between objects. I assume that a 3D semantic graph map $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is generated in prior. This graph $\mathcal{G}$ consists of nodes $v_i \in \mathcal{V}$ and edges $(v_i, v_j) \in \mathcal{E}_{ij}$. A node feature vector X^i of v_i is composed of RGB image features $F^i_{RGB} = \{f_1, \ldots, f_n\}$, depth image features $F^i_D = \{f_{n+1}, \ldots, f_{n+m}\}$, and position $P^i = \{p_1, \ldots, p_n\}$ of an object in the RGB-D image.

Bounding boxes of objects can be obtained using segmented images from publicly available dataset or object detection algorithms. The RGB image of an object cropped along the bounding box is converted to an RGB feature F^i_{RGB} through a pre-trained neural network. These RGB features can have semantic meanings such as object category. Similar to F^i_{RGB}, the depth image of an object cropped along the bounding box is transformed into depth features F^i_D encoding the 3D shape through CAE. The position vector P^i is defined as the center of the bounding box. By adding the position vector as the element of graph node v_i, the correlation between the positions of the objects in a 3D scene can be considered when performing the graph feature extraction.

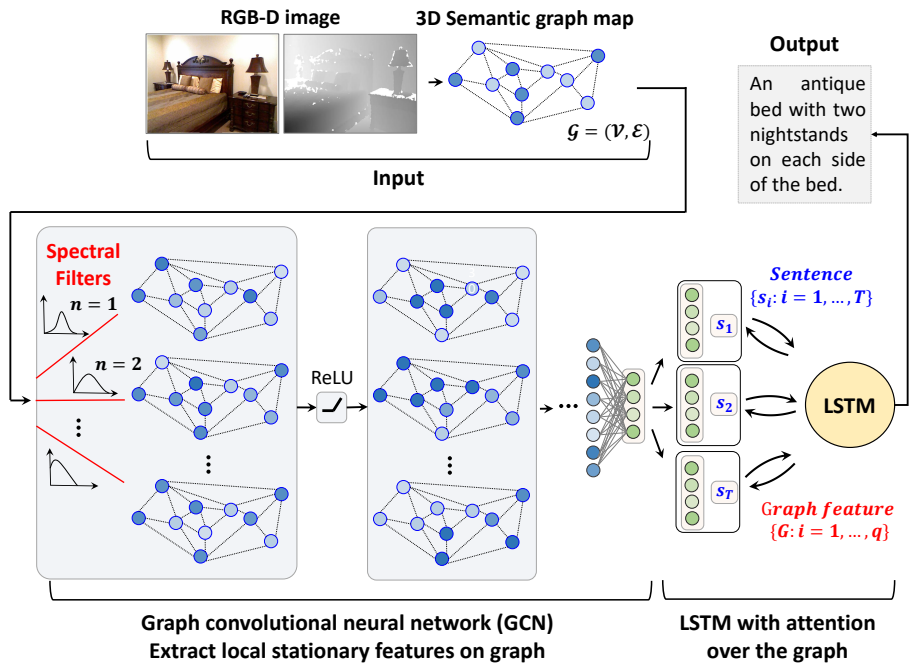


Figure 3.3: *Overview of the natural language description based on the semantic graph map: A 3D semantic graph map is generated using an RGB-D image. Features of a graph are extracted using GCN which performs spectral graph convolution operation defined by spectral graph theory. Then, LSTM generates a sentence with focused over the graphs* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*. Inputs of LSTM are a graph feature vector* G *and a sentence* $\{s_t : i = 1, ..., T\}$*. I train this network using graphs generated by a single scene. Unlike other methods, I can generate a sentence using a graph for not only a single scene but also multiple scenes.*

3.3.2 Graph Feature Extraction

This section describes a method of natural language sentence generation using a 3D semantic graph map. The key difference of the proposed method compared to conventional methods is that multiple scenes can be described even though a network is trained only using a single scene. Since my approach is based on a graph, it has an advantage that multiple scenes can be implemented as a single graph. Therefore, a graph with different amount of nodes can be used for generating descriptions. In addition, I can generate consistent sentences regardless of node ordering due to the characteristic of a graph. The overall network architecture of my method is shown in Figure 3.3. GCN extracts features of nodes and edges, and RNN generates a sentence describing the graph. I explain each step in detail as follows.

Since convolution operation for convolutional neural network (CNN) is only defined for a regular grid, the graph structure of irregular and non-Euclidean form is hardly applied [52]. On the other hand, GCN which performs a convolution operation on the graph defined by spectral graph theory can be utilized for a graph map feature extraction. Usually, the computational complexity of spectral graph convolutions becomes very high [53]. Therefore, I reduce learning complexity by following a localized first-order approximation of spectral graph convolutions [54] to encode the structure of an object-oriented 3D semantic graph map. $A \in \mathbb{R}^{N \times N}$ (binary or weighted) is an adjacency matrix of the graph indicating the pair-wise relation among objects, and $D_{ii} = \sum_j A_{ij}$ is the degree matrix. Binary is used to indicate the connectivity of edges. The weighted value is used to indicate relations of each object with a specific value such as Euclidean distance between two nodes. The process of layer-wise propagation of a spectral graph convolution is as following:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W_g^{(l)}) \tag{3.1}$$

$\hat{A} = A + I$ is an adjacency matrix with self-connection added, where $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$. $W_g^{(l)}$ is a learnable variable and σ is an activation function. $H^{(l)} \in \mathbb{R}^{M \times D}$ is the output

from l-th layer, where $H^{(0)} = X$. The learning process of k spectral graph convolution layers is as following:

$$f(X, A) = \sigma(\tilde{A} \dots \sigma(\tilde{A} X W_g^{(0)}) \dots W_g^{(k)}) \tag{3.2}$$

where inputs are $X = \{X_1, ..., X_N\}$ whose elements are the features of nodes. $\tilde{A} = \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$ is computed in the preprocess. The output of the GCN is encoded as q-dimensional vectors $G = \{g_i : i = 1, \dots, q\}$ by fully-connected layer as following:

$$G = \sigma(W_m(f(X, A)) + b_m) \tag{3.3}$$

W_m, b_m is learnable parameters. Through this process, I can extract local stationary features of an object-oriented 3D semantic graph map composed of nodes and edges.

3.3.3 Natural Language Description with Graph Features

I generate a natural language sentence using RNN with graph features obtained by GCN. The inputs of RNN are a vector concatenated with a word vector s_t and a graph feature vector G. This network calculates the hidden state h_t and an output word vector o_t according to $t = 1, \dots, T$. I closely follow long short-term memory (LSTM) [45] which is a type of RNN:

$$\begin{aligned}
u_t &= \{h_{t-1}; g_t; s_t\} \\
i_t &= \sigma(W_i \cdot u + b_i) \\
f_t &= \sigma(W_f \cdot u + b_f) \\
o_t &= \sigma(W_o \cdot u + b_o) \\
c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot x + b_c) \\
h_t &= o_t \odot tanh(c_t)
\end{aligned} \tag{3.4}$$

i_t, f_t, o_t, c_t, h_t are input, forget, output, memory, hidden state of LSTM, respectively. W_i, W_f, W_o, W_c and b_i, b_f, b_o, b_c are learnable parameters.

LSTM training

LSTM is trained to combine an input vector u_t and a context vector h_{t-1} from the previous layer from the previous layer to predict the probabilistic distribution over the next word. To generate a sentence, given an object-oriented 3D semantic graph map, LSTM is conditioned on the graph information G for every step. The learning process is as follows. I initialize h_0 as $\vec{0}$, x_1 as a start token and a target label o_1 to the first word in a sentence. Similarly, I set x_2 as the first word vector, then LSTM is expected to predict the probability distribution of the second word o_2. This process is repeated until the last word and desired label are set as x_T and an end token. A graph feature vector G is applied along with x_t, h_t through all the learning process.

LSTM at test time

I can generate a sentence describing both a single scene and multiple scenes using the LSTM, which is trained only with a single scene. To predict a sentence, a graph feature vector G is computed using a graph consisting of a single scene or multiple scenes. Then, h_0 as $\vec{0}$, x_1 as a start token with G is feed into the network predicting a next word o_1. Analogously, o_1 is applied as x_2 with G. This process is repeated until an end token occurs.

3.4 Experiments

Datasets

To evaluate the performance of the proposed method which generates an object-oriented description for both a single scene and multiple scenes, experiments were performed using two datasets. One is NYUv2 Sentence dataset [44] consisting of single scenes. The other is SUN 360 panorama dataset [55] composed of multiple scenes. I use NYUv2 Sentence dataset which contains 1,449 RGB-D images, segmented images, and $3 \sim 5$ sentences for training.

Preprocess

To perform image augmentation, I randomly applied salt and pepper noise or affine transform. An object-oriented 3D semantic graph map is generated using RGB-D and segmented images. The node information of the graph I used is RGB-D image features and the position of the bounding box. RGB image features are extracted using VGGNet [51], as a 4096-dimensional vector. Depth image features are extracted using CAE as a 64-dimensional vector. For objects in a single image, I connect objects that are close to each other using Delaunay triangulation. For multiple images, I extend the graph by connecting the overlapping objects among images. To cover the dynamic number of objects, I set the maximum number of nodes to 80, which can cover at least $10 \sim 12$ scenes for SUN 360 dataset. If the number of objects n in an image are less than 80, I set $X_i = \vec{0}\ \forall i > n$. For these empty nodes which are assigned as $\vec{0}$, I assume that they are not connected to other nodes but itself. For the single scene with the small number of nodes, some of the elements of X are almost always zero when empty nodes are gathered to one side of X. Therefore, the network trained with the graph of the single scene learns to discard some parts of the element in X. As a result, a graph composed of multiple scenes has not 0 vectors but a large number of nodes. Thus it is difficult to generate a sentence considering the entire nodes. To relax this problem, I shuffled the sequence X as well as A, including empty nodes. I also randomly disconnect the edges of the graph to make it possible to deal with semantic maps generated from the various graph SLAM algorithms.

Training networks

The overall architecture of proposed method consists of three parts as shown in Table 3.1. The first part extracts features from a graph with two graph convolution layers and one fully connected layer. As shown in Table 3.1, the graph convolution layer needs two inputs, one is node feature vectors and the other is the adjacency matrix of edges. The second part learns the sentence structure itself using embedding layer and

LSTM layer. The final part generates a sentence that describes graph using LSTM and fully-connected layer. Details of input and output for each layer are shown in Table 3.1. All parameters were learned using the Nadam [56], and relu was used as an activation function in each layer except the last layer. In the last layer, a softmax classifier is used to output prediction probabilities. Also, dropout is added by 0.5 after each layer to reduce overfitting.

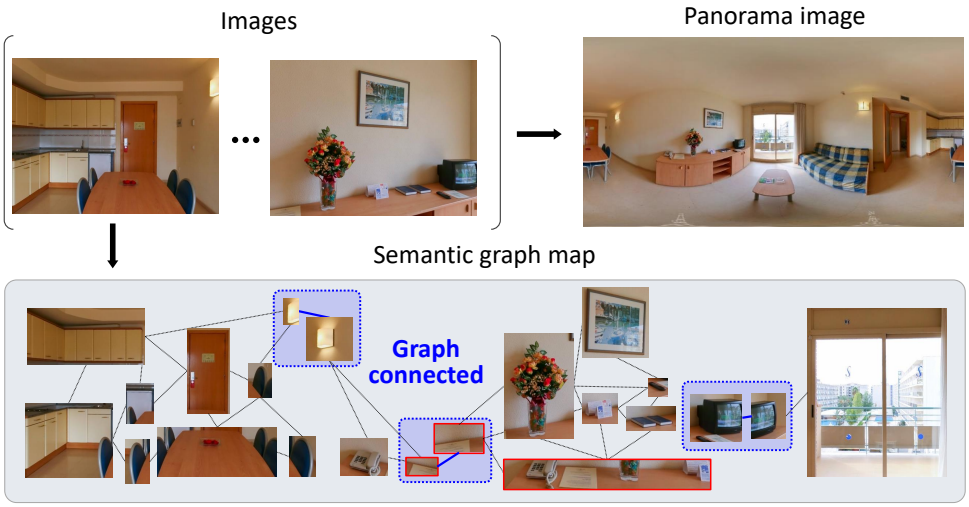
Table 3.1: Architecture of trained neural network for language description

Layer type	Filters/ Units	Output size	Number of parameters
Input(Node)	-	80×4163	-
Input(Edge)	-	80×80	-
Graph convolution1	1024	80×1024	$4,262,912$
Graph convolution2	64	80×64	$65,536$
Fully Connected1	-	512	$2,621,952$
Input(Words)	-	44	-
Embedding	256	44×256	$798,976$
LSTM1	256	44×256	$525,312$
LSTM2	1000	1000	$7,076,000$
Fully Connected2	-	3121	$3,124,121$

Table 3.2: Performance of the proposed and comparison methods

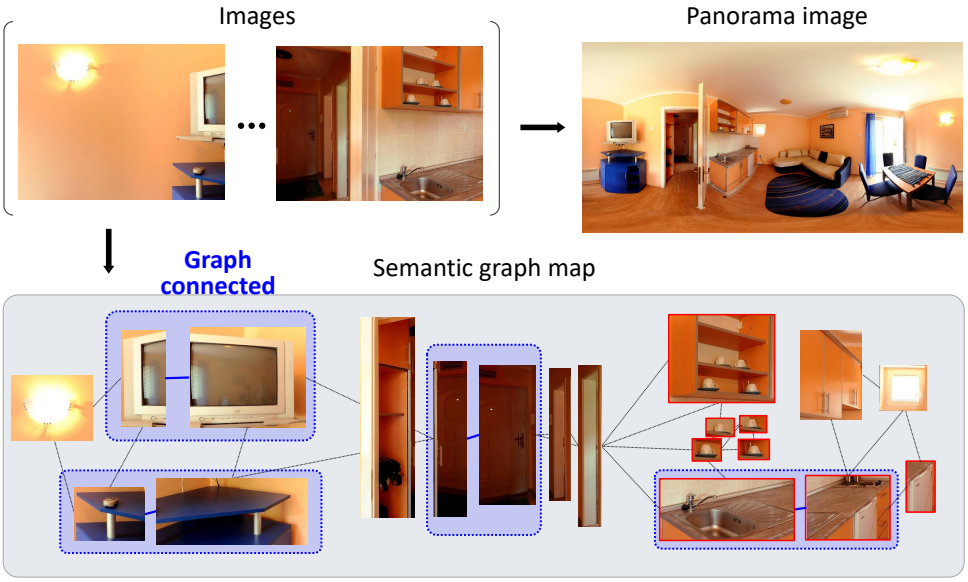
Model	BLEU
Vinyals et al. [45] (RGB)	43.97
Karpathy et al.[46] (RGB)	27.54
Vinyals et al. [45] (RGB-D)	45.04
Proposed model (semantic graph)	43.41

Figure 3.4: *Example sentences generated using my model (single scene): A semantic object-oriented graph map is generated using objects that are detected by bounding boxes. A node contains RGB-D image features and bounding box information of an object. Edges are all connected since they are all in the same scene. I colored the object boxes red and underlined the nouns referring to them in the sentence.*



Panorama image: This is a picture of a living room.

Semantic graph map: A few pieces of paper are on the table in the front.



Panorama image: This is a picture of a living room.

Semantic graph map: There is also a dishwasher below a sink and a few other items on top of the counter.

Figure 3.5: *Example sentences generated using my model compared to a conventional method (multiple scenes).*

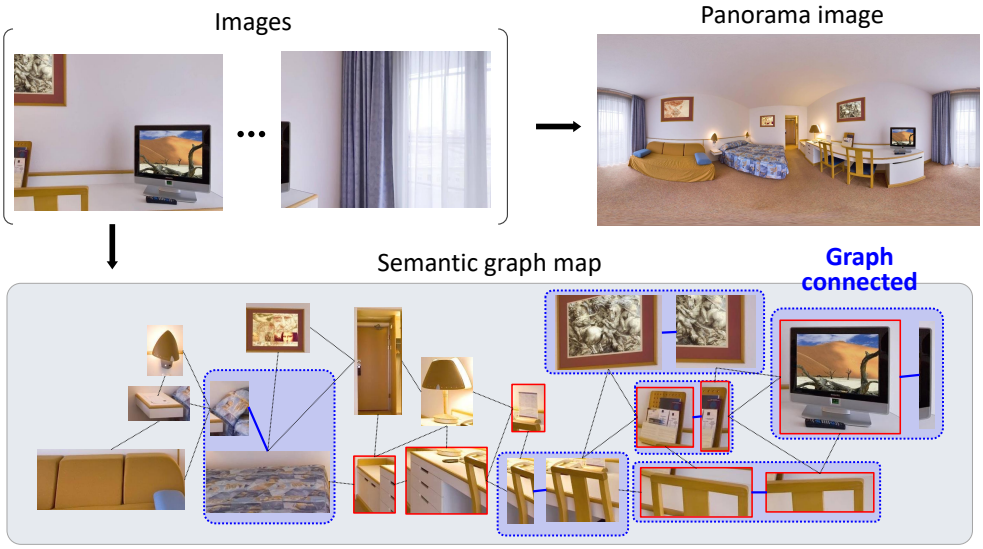


Figure 3.6: *Example sentences generated using my model compared to a conventional method (multiple scenes): To construct a graph using multiple scenes, overlapped objects are used as a common node. I compared my model with [45] using panorama images. The results show that my model utilized objects in multiple scenes more efficiently than conventional method by generating an object-oriented sentence. I colored the object boxes and underlined the nouns referring to them in the sentence. The blue boxes show the connected parts of the graph.*

Scene description

I validate the proposed method using datasets consisting of a single scene or multiple scenes. BLEU [57] is used as an evaluation index compared to other methods [45], [46]. However, in order to evaluate the quality of sentences using the BLEU, reference sentences are required. Since the NYUv2 dataset consists of a single scene image and corresponding sentences, the generated sentences can be evaluated using BLEU. On the other hand, the experimental results generated using the SUN 360 panorama dataset composed of multiple scenes are difficult to evaluate because there are no reference sentences. Therefore, I only use BLEU as an evaluation index for natural language description generated using a single scene.

Table 3.2 shows the quality performance of generated sentences using the network architecture in Table 3.1. For both Vinyals et al. [45] and Karpathy et al. [46], I used the pre-trained neural network VGGNet to extract RGB image features. Vinyals et al. [13] uses the RGB image feature as the input to LSTM, whereas, Karpathy et al. [45] uses the extracted feature as the bias in the RNN with the first word in the sentence. Since conventional methods hardly utilize 3D information such as RGB-D images, I modified the network architecture of [45] into the form that can make use of the RGB-D images. I applied the depth features extracted by CAE as the new input, reduced the dimension using a fully connected layer, and concatenated the depth features with RGB features. The rest of process to generate sentence describing an image is the same as the existing method. As a result, the BLEU score is slightly improved compared with the case of using only RGB features. Examples of predicted sentences are shown in Figure 3.4. I can verify that the sentences were generated with attention over objects in the scene. Moreover, I tested the network using a graph with different ordering of nodes for the same single scene, and verified that the same sentence is always generated due to the characteristic of a graph.

The SUN 360 panorama dataset consists of sequential images and corresponding panorama images are used to demonstrate that my approach generates a sentence over

multiple scenes focusing on objects. Figure 3.5 and Figure 3.6 shows example sentences generated using this dataset by the proposed method compared to [45], which showed the best performance on a single scene. To construct an object-orient graph using multiple images, overlapped objects from each image are used as common nodes. Panorama image which is representing overall scene is utilized for [45] to generate a natural language sentence. As shown in Figure 3.5 and Figure 3.6, the existing method generated the same sentence for different images. On the other hand, my model generated a more precise sentence focusing on objects for multiple scenes, even though the quality of the sentence was slightly degraded. Therefore, I can conclude that the proposed algorithm successfully generates an object-oriented description over 3D semantic graph map composed of both a single scene and multiple scenes.

3.5 Summary

A new working environment understanding method using natural language is introduced for human-robot communication. In order to efficiently utilize 3D information and multiple scenes, I generate a natural language description using an object-oriented 3D semantic graph map. The graph map is constructed using RGB-D images. Nodes of this graph contain features and position of detected objects, and edges show pair-wise relationships between objects. GCN and LSTM are used to generate a sentence regarding the semantic graph. I evaluated the performance of the proposed method compared with existing algorithms using NYUv2 Sentence and SUN 360 panorama datasets. As a result, I verified that my algorithm successfully generated a sentence for both a single scene and multiple scenes using 3D information. Moreover, the proposed approach generated a sentence describing objects more precisely than conventional methods due to the object-oriented graph characteristic. Since the proposed method is focusing on generating one sentence for multiple scenes, future work concerns with generating multiple sentences for an object-oriented 3D semantic graph map understanding.

Chapter 4

Natural Question with Semantic Graph

4.1 Introduction

In robotics, natural question generation is necessary for autonomous robots to formulate problems [58].In fact, a truly autonomous robot must be able to propose problems according to its surroundings and find solutions from its perception [59, 60]. Unlike most studies on solving problems with minimal time and effort, research on natural question generation are scarce. An inverse semantics algorithm is proposed in [2] to generate questions when a robot encounters unexpected circumstances, enabling it to reset problems according to the environment. However, the practical application of this method is limited because questions are generated only for designed environments. To handle a variety of situations, learning techniques applied to computer vision are used to generate natural language captions based on images [61, 46]. However, these techniques cannot be applied to unstructured data. In contrast, semantic graphs are widely exploited by robots to understand their surroundings. Moreover, as most existing methods can only process a single image at a time, it is difficult to generate natural questions for sequential scenes that represent more realistic working environments of robots.

In this dissertation, I propose a method for generating natural questions using object-oriented semantic graphs. Figure 4.1 shows a graph with features of detected

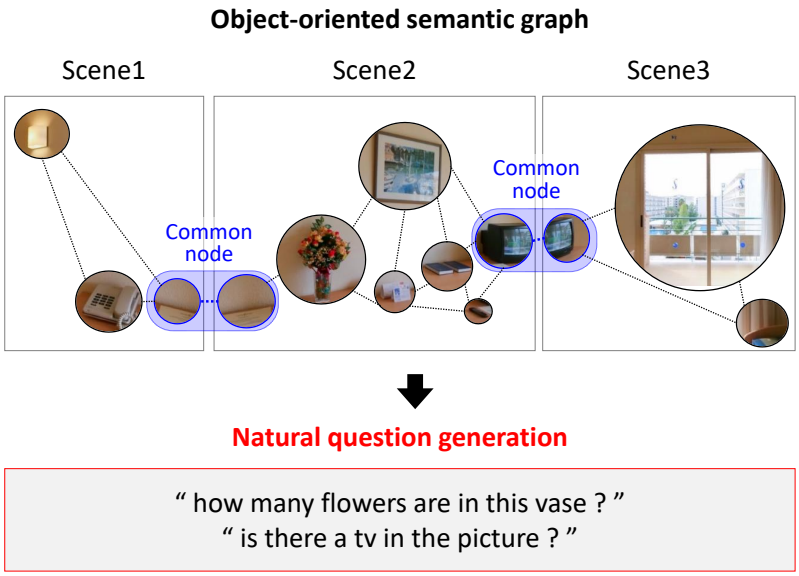


Figure 4.1: *Natural question generation using object-oriented semantic graph that can represent sequential scenes: A semantic graph is constructed using overlapped objects from multiple scenes as common nodes. Then, natural questions are generated with attention over a semantic graph*

objects in an environment as nodes and their relationships as edges. Therefore, a semantic graph can represent information of objects over sequential scenes. By using a graph convolutional network (GCN), features are extracted from object-oriented semantic graphs, whose structure is irregular and non-Euclidean [54]. Then, a recurrent neural network (RNN) considers the graph features and corresponding captions as input for training to generate natural questions. I verify the proposed method on publicly available datasets.

4.2 Related Work

This dissertation is part of the three flows of studies: Artificial intelligence for autonomous robots, graph neural network (GNN), and image based natural question generation.

Artificial intelligence for autonomous robots

Autonomous agents should not be simple passive observers but need to act actively. [62]. In other words, robots must be able to formulate problems by performing natural question generation and solve them. [58] combined the questioning phase of active learning with learning from demonstration techniques to improve the work efficiency of a robot. [63] performed "I Spy" game, which learns natural language words by describing in turns between a robot and human user. [64] solved the multi-robot remote driving problem by proposing a system model with a dialogue, asking questions and answering, between robots and humans. [65] presents a natural question generation method for human-robot collaborative tasks. However, these studies can only generate questions related to the designed environment. I propose a method capable of generating questions for various visual information.

Graph Neural Network

Data-driven deep learning has enabled rapid progress in image classification, natural language process, and video process. In particular, convolutional neural network (CNN), which extracts features from images and learns the pattern, and RNN, which is specialized for sequential data, are widely used in applications such as machine translation and autonomous navigation [66]. However, because these neural networks can only learn information from Euclidean space, they cannot be applied for none-Euclidean data such as graphs. Irregular forms of data that can be represented as graphs are widely used in various areas such as chemistry, knowledge bases, and natural language. Specifically, a graph-based learning system, which exploits the relationship between users and products, showed good performance in e-commerce [67]. Many researchers have proposed GCN [68], graph spatial-temporal networks [69], graph auto-encoders [70], and graph generative networks [71] to make use of graph data in deep learning. [72] demonstrated graph feature learning capabilities of gated graph neural networks via experiments on bAbI tasks [73]. [54] proposed a semi-supervised learning method for GCN, which can learn not only labeled data but also unlabeled data. [53] and [74] performed 3D point cloud and image classification using spectral based GCN, respectively. [48] utilized GNN to predict semantic meanings of images by finding corresponding verbs. [16] generated a scene graph with annotations of object, phrase, and region for an image through a single neural network model. In this study, I generated natural questions for a truly autonomous robot with a spectral based GCN.

Image Based Natural Question Generation

Image based natural language processing is a very important problem in the artificial intelligence perspective of robots. Many researchers conducted studies on natural language sentence generation with visual information by combining computer vision and natural language process [62]. [46] found inter-modal correspondences between image

regions and natural sentences using region-based CNN and bidirectional RNN. [75] described images by finding stronger semantic contents with images and annotations. Recently, visual question answering as well as image captioning attracted wide attention [76]. [77] proposed an end-to-end neural network without intermediate stages such as image segmentation and object detection to find an image-based answer for simple questions. However, these methods have a limitation in that they rely on manually constructed questions.

Some studies on learning to ask questions about a single image have been researched. [78] proposed an image based automatic question-answer system using uncertainty of Bayesian framework and image segmentation techniques. [79] constructed a framework for object proposal, unknown object identification, and visual question generation and achieved information about unknown information directly from humans. [76] generated various types of questions for a single image through a model consisting of CNN, RNN, and question type selectors. In a similar manner, [80] generated visual grounded questions using three different generative models. Unlike these researches, I discuss an algorithm that can generate natural questions related to multiple images as well as a single image.

4.3 Natural Question Generation

The key difference between existing algorithms and the proposed method from the computer vision perspective is that natural questions are generated using object-oriented semantic graphs rather than images. A semantic graph is a map representation method, which is widely studied in robotics [8, 11]. Unlike conventional map representation that simply shows the status of the environment using scatter, line, or surface, this graph contains information about the types of objects and spatial relationship between objects. Robots utilize the rich information of semantic graphs when performing localization, path planning, and collision avoidance [81]. This applicability has boosted

research on semantic mapping. I use semantic graphs to generate natural questions.

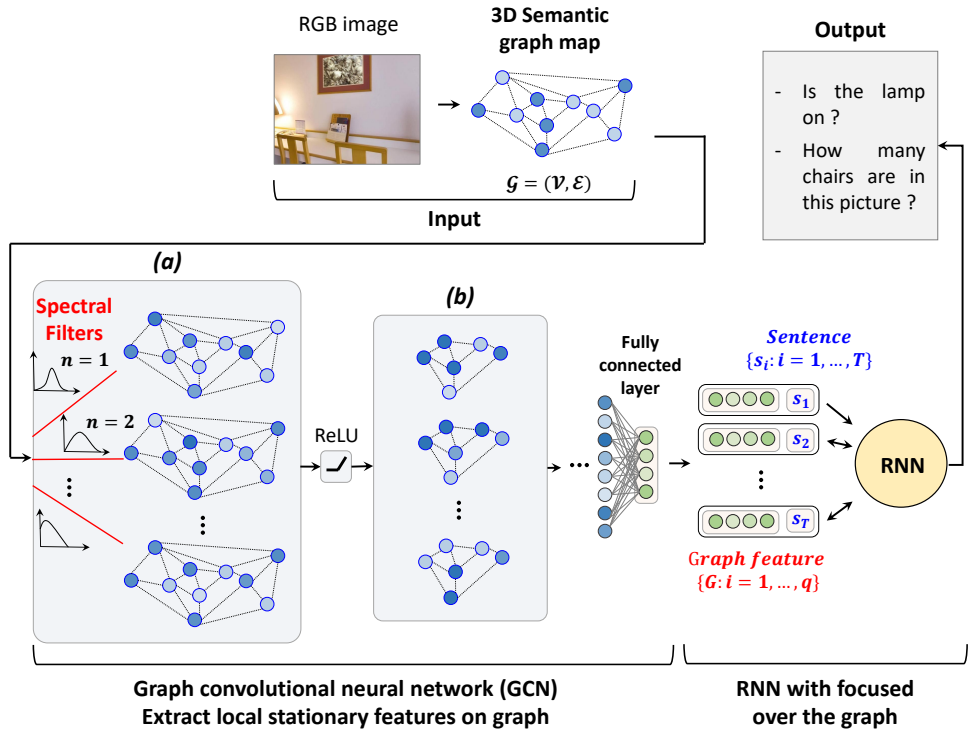


Figure 4.2: *Graph feature is extracted by GCN. LSTM takes a word concatenated with a graph feature as input and generates natural questions. GCN architecture comprising spectral graph convolution layers, pooling layers, and fully connected layers applied to semantic graph $\mathcal{G}$: (a) Spectral graph convolution layer (b) Graph pooling layer consisting of graph coarsening and max-pooling operation.*

This section describes a natural question generation method based on object oriented semantic graphs. The overall process is shown in Figure 4.1. I assume that the semantic graphs are generated in advance. After extracting features from the graphs by GCN, natural questions are generated using RNN. The architecture of the proposed neural network is illustrated in Figure 4.2. Because the proposed algorithm generates natural questions with graphs, it has an advantage in that I can test not only a single

scene but also sequential scenes using a neural network trained with graphs constructed with a single scene.

4.3.1 Preprocess

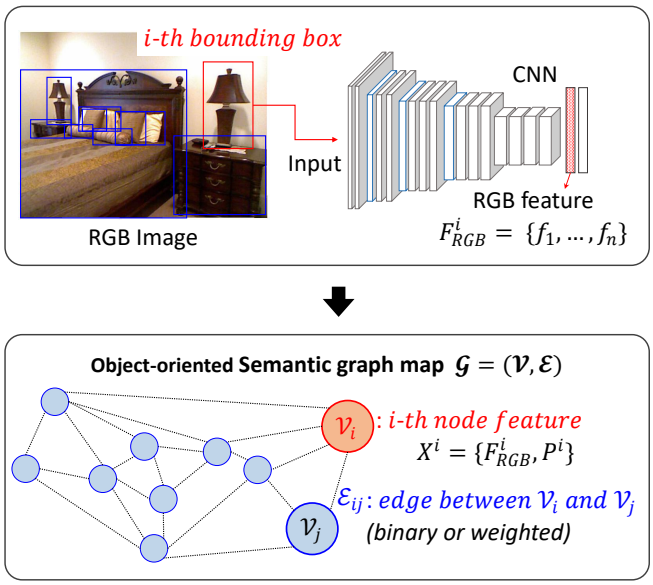


Figure 4.3: *A semantic graph map $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed using features F^i_{RGB} and position P^i of detected objects. F^i_{RGB} is extracted by a pre-trained CNN. An edge $\mathcal{E}_{ij}$ denotes the relation between two objects.*

Semantic graph mapping using graphical models has been commonly used for map generation in robotics. This map has meaningful information combining semantic and geometric data. During SLAM, semantic information is used for data association in the front-end and geometric information is utilized for graph optimization in the back-end. I use this rich structure to generate natural questions. In this study, graphs are constructed as similar to the graph created by semantic SLAM.

Unlike existing algorithms, which use a single image for natural question generation, a semantic graph can efficiently represent sequential scenes by increasing the

number of nodes and edges. Nodes $v_i \in \mathcal{V}$ of graph map $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent objects and the edges $(v_i, v_j) \in \mathcal{E}_{ij}$ represent the relationships among the nodes. Feature vector X^i of node v_i comprises position $P^i = \{p_1, \ldots, p_n\}$ and features $F^i_{RGB} = \{f_1, \ldots, f_n\}$ of an object. In this study, I used RGB image features that can be obtained from a pre-trained neural network as illustrated in Figure 4.3. A 4096-dimensional vector extracted using VGGNet [51] and a 2-dimensional position vector composed of x and y of an image are concatenated, and then used as node information. I use an unweighted graph that only shows whether an edge is connected or not, with 0 and 1. For sequential scenes, I consider overlapped objects as common nodes.

4.3.2 Graph Feature Extraction

Although CNN can extract features in high-dimensional and large-scale datasets, they cannot be applied for irregular and non-Euclidean graphs because they are only defined for regular grids. Instead, I use a GCN [53] composed of graph convolution layer, pooling layer, and fully connected layer as shown in Figure 4.2. To encode the structure of an object-oriented semantic graph, graph convolution defined by spectral graph theory is performed as follows:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W_g^{(l)}), \tag{4.1}$$

$\hat{A} = A + I$ is an adjacency matrix $A \in \mathbb{R}^{N\times N}$ (binary or weighted) with self-connection I, where $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ is a degree matrix, $W_g^{(l)}$ is a trainable variable and $\sigma(\cdot)$ is an activation function, such as rectified linear unit (ReLU). The output of the l-th layer is $H^{(l)} \in \mathbb{R}^{M\times D}$, where $H^{(0)} = X$. My model $f(X, A)$ with k spectral graph convolution layers is given by

$$f(X, A) = \sigma(\tilde{A}\ldots\sigma(\tilde{A}XW_g^{(0)})\ldots W_g^{(k)}), \tag{4.2}$$

where $\tilde{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$. The inputs are a node feature vector $X = \{X^1, \ldots, X^N\}$ and an adjacency matrix A of graph $\mathcal{G}$. To reduce the high computational complexity of spectral graph convolutions, I adopt their first-order approximation.

Pooling layers perform graph coarsening [82] and max-pooling to down-sample the graph while preserving meaningful nodes and edges. First, I use geometric information P^i of v_i for graph coarsening. Then, max-pooling is performed using the output of a graph convolution layer. Finally, a fully connected layer encodes graph convolution features as q-dimensional vector g as follows:

$$g = \sigma(W_m(f(X, A)) + b_m), \tag{4.3}$$

where W_m and b_m are trainable parameters. I train the GCN using a semantic graph constructed for a single scene. During test, I extract features from a graph consisting not only of a single but also sequential scenes unlike conventional algorithms.

4.3.3 Natural Question with Graph Features

I generate natural questions based on a long short-term memory (LSTM), which is a type of RNN. My implementation of LSTM closely follows [13]. I propose a simple but effective extension that can additionally condition the question generation process on an input graph. My model takes a word concatenated with a graph feature vector as input. Then, the next word is computed to fit the graph. When training LSTM, I also back-propagate GCN expecting to extract appropriate graph feature for question generation.

Question generation aims to predict the probabilistic distribution over target word vector o_t, by combining input vector u_t and hidden vector h_{t-1} with $t = 1, \ldots, T$. Concatenated vector u_t is composed of g and a word vector s_t. Therefore, my model generates natural questions conditioned on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for both a single scene and sequential scenes by using g, which represents features of an object-oriented semantic graph. For initialization, I set s_1 as a start token, h_0 as $\vec{0}$, and o_1 as the first word in a sentence. Analogously, s_2 is set as the first word and o_2 is set as the second word. This process ends when o_t is set as an end token. The training variables of the network are updated using stochastic gradient descent with adaptive learning rate methods, and a

ReLU is the activation function for each layer except for the last one. Then, a Softmax classifier predicts probabilities for the last layer. The network is decoded using a beam search of size 6. I compose the vocabulary list of a training dataset with words that are seen more than three times, and thus I set filtered words as an unknown token. During test, I prevented the unknown token to be produced.

4.4 Experiments

Because there is no dataset for question generation using graphs, I used three datasets to validate the proposed method, namely, the VQA dataset [83] composed of images, object boundary boxes, and captions, the SUN360 panorama [55], and the Stanford 2D-3D-semantics datasets [84] consisting of indoor sequential scenes. I generated a semantic graph before using the object boundary boxes provided with the VQA dataset. F^i_{RGB} was extracted as a 4096-dimensional vector using a pre-trained network for every v_i. Delaunay triangulation was used for edge connection $\mathcal{E}_{ij}$ between v_i and v_j based on the distance between objects. To deal with the varying size of graphs, I set the maximum number of nodes to 20, which can cover 3-5 sequential scenes of the SUN360 panorama and the Stanford 2D-3D-semantics datasets. For less than 20 nodes, I added empty nodes only connected to themselves that vanished after graph pooling during training. In addition, I randomly disconnected graphs to cope with various semantic mapping algorithms when training. The architecture of my network consists of two parts, namely, graph feature extraction and question generation. First, the GCN extracts feature vectors from the graphs, and then the embedding layer and LSTM learn to generate questions for the object-oriented semantic graph. The network architecture is detailed in Table 4.1.

Six different evaluation metrics: BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, and ROUGE were adopted for quality evaluation of the generated questions. Given that these metrics require reference sentences for evaluation, evaluation of the SUN360

Table 4.1: Architecture of trained neural network for natural question generation

Layers	Filter	Shape	Connected to
Input (Node)	-	20×4098	-
Input (Edge)	-	20×20	-
Graph conv 1	2048	20×2048	Input (Node) Input (Edge)
Graph conv 2	2048	20×2048	Graph conv 1
Graph conv 3	2048	20×2048	Graph conv 2
Graph pool 1	-	15×2048	Graph conv 3
Graph conv 4	1024	15×1024	Graph pool 1
Graph conv 5	1024	15×1024	Graph conv 4
fc1	-	4096	Graph conv 5
Input (Words)	-	15	-
Embedding	512	15×512	Input (Words)
LSTM 1	1024	15×1024	fc1 Embedding
fc2	-	5521	LSTM 1

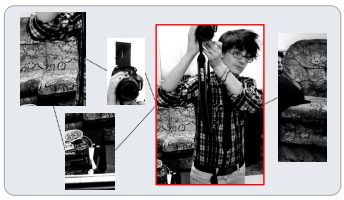
Table 4.2: Performance of the proposed and comparison methods

Method	Xu et al. [13]	Mostafazadeh et al. [80]	The proposed model
BLEU-1	61.7	35.1	70.5
BLEU-2	42.2	21	51.1
BLEU-3	30.2	16.3	33
BLEU-4	22.7	13.5	22.1
METEOR	19.98	11.8	23.8
ROUGE	50	31.9	55.4

Semantic graph

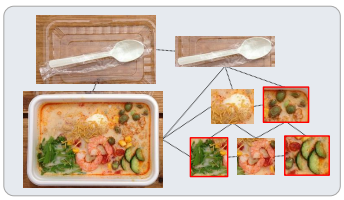
Xu et al.	what color is the bench ?
Mostafazadeh et al.	is the bench shiny ?
Our model	what is the bench made of ?

Semantic graph



Xu et al.	is this person wearing glasses ?
Mostafazadeh et al.	is the man outside ?
Our model	what is the man holding ?

Semantic graph

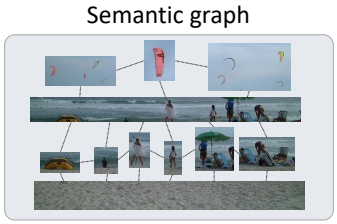


Xu et al.	what color is the plate ?
Mostafazadeh et al.	is the salad fresh ?
Our model	what is the green vegetable ?

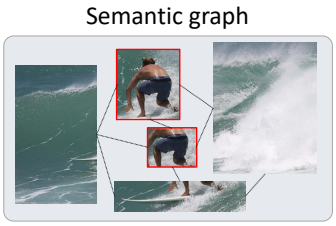
Semantic graph

Xu et al.	what room is this ?
Mostafazadeh et al.	is that a bathroom ?
Our model	what is the color of the wall ?

Figure 4.4: *Examples of generated questions from object-oriented semantic graphs (single scene): I colored the object boxes and underlined the nouns referring to them in the questions.*



Xu et al.	what is in the sky ?
Mostafazadeh et al.	how many kites are there ?
Our model	what is flying in the air ?



Xu et al.	what is the man doing ?
Mostafazadeh et al.	is the man getting wet ?
Our model	is the man wearing a wetsuit ?

Figure 4.5: *Examples of generated questions from object-oriented semantic graphs (single scene): I colored the object boxes and underlined the nouns referring to them in the questions.*

panorama and the Stanford 2D-3D-semantics datasets is difficult because it only contains images. Therefore, I only evaluated these metrics on the VQA dataset, obtaining the performance listed in Table 4.2. For comparison, I implemented the methods in [13] and [80], where the former uses a LSTM with the attention model, and the latter retrieves natural questions using a generation model based on a multimodal RNN. As the method in [13] is intended to generate natural language descriptions of images, I adapted it to generate natural questions. My method outperforms the others in most of evaluated metric. Even though the score of BLEU-4 is slightly degraded, examples of generated questions for a single scene show that the proposed method successfully generates questions over detected objects and their attributes such as 'green vegetables' as illustrated in Figure 4.4 and Figure 4.5.

Existing methods such as those in [13] and [80] have rarely been applied to multiple scenes because they are based on single-image processing. In contrast, my method can generate questions for sequential scenes by using overlapping objects as common

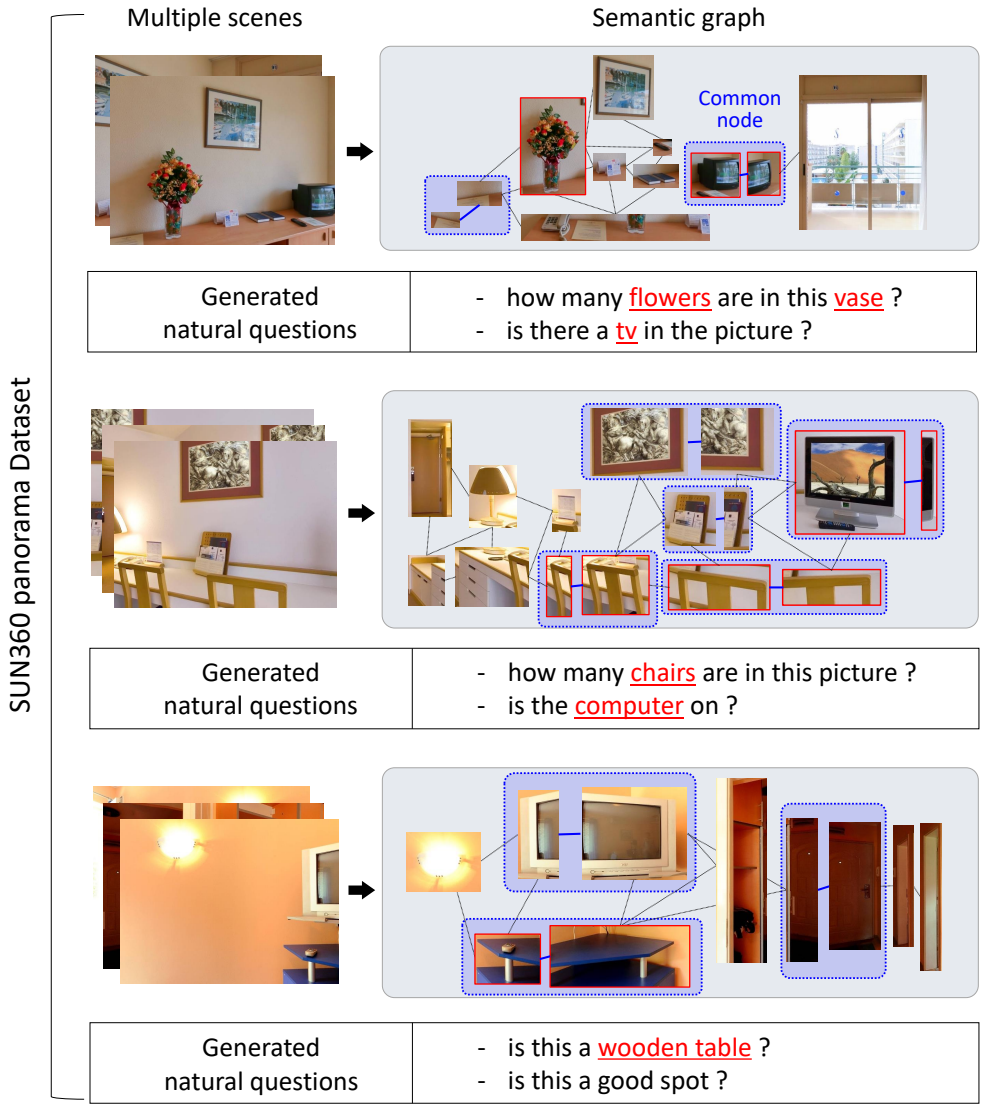


Figure 4.6: *Examples of generated questions from object-oriented semantic graphs (multiple scenes)*

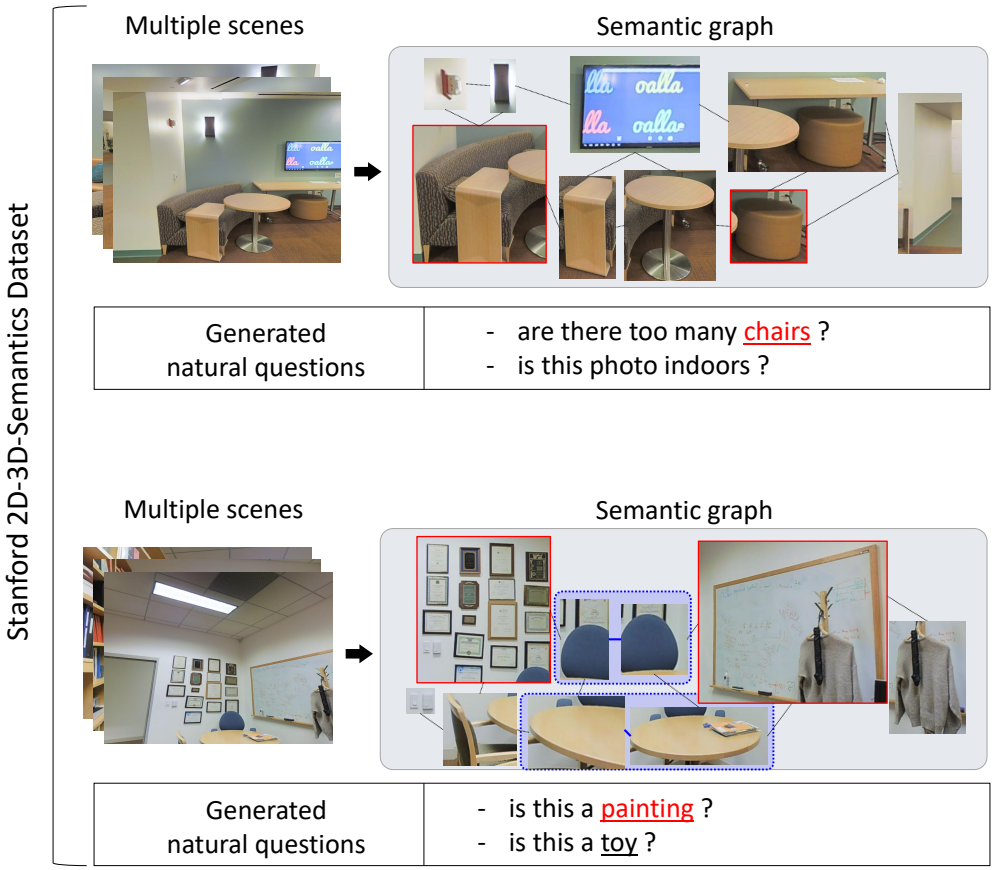


Figure 4.7: *Examples of generated questions from object-oriented semantic graphs (multiple scenes): For the same graph, I shuffled the sequence of nodes and edges. As a result, my model generates different sentences focusing on different objects. I colored the object boxes and underlined the nouns referring to them in the questions. The blue boxes show the connected parts between scenes.*

nodes. I used the SUN360 panorama and the Stanford 2D-3D-semantics datasets to demonstrate the ability of the proposed method on multiple scenes. Examples of generated graphs and the natural questions are shown in Figure 4.6 and Figure 4.7. I generated several questions based on the same graph but shuffling the sequence of nodes and edges. Consequently, the method generated different questions for the objects depending on the sequence. Generated questions can be divided into two categories. One is questions about objects and the other is questions about whole scenes. I can conclude that the proposed method successfully generates diverse natural questions over semantic graphs comprising either single or sequential scenes, which are more realistic for the working environments of robots.

4.5 Summary

I propose a method for generating natural questions from semantic graph mapping aiming to enhance robot autonomy. I use object-oriented semantic graphs resulting from graph mapping for question generation. To extract graph features, a GCN performs a convolution on graphs. Then, an RNN generates the natural questions. From the proposed method, graphs consisting of a single or sequential scenes can be used for natural question generation. Experiments on the VQA, SUN360 panorama and Stanford 2D-3D-semantics datasets verify that the proposed method successfully generates natural questions for both single and sequential scenes, outperforming existing methods.

Chapter 5

PDDL Planning with Natural Language

5.1 Introduction

Automated planning, also known as artificial intelligence (AI) planning, has always been an important component of research on AI. Automated planning generates a series of actions sequentially to achieve the desired goal state from the initial state while satisfying certain constraints. It is employed in various robotics applications, ranging from autonomous manufacturing to generating dialogue agents. Automatic planning is categorized into offline planners and online planners according to the ability to cope with dynamic environments. In the field of robotics, the use of the online planner is inevitable in that robots are exposed to continuously changing situations. Many kinds of online planning studies have been conducted using various sensors [85, 86]. They transmit raw sensor data in real time to perceive surrounding environments and achieve the planning goal. However, it is difficult to cope with mission failure due to various situation changes with raw sensor data alone. Moreover, traditional automated planning methods are limited with regard to domain model acquisition [87], as shown in Figure 5.1. They assume that users can formulate the initial state and the corresponding behaviors of the robot working space using formal language. In order to alleviate these problems, this dissertation proposes a method that enables robots to cope with various

situations by utilizing natural language sentences provided by humans. ROSPlan [88], a planner for robots to perform mission planning, and a natural language grounding technology that enables robots to automatically express informal human natural language sentences into a formalized form are utilized. Experiments are conducted for two categories of scenarios for validation.

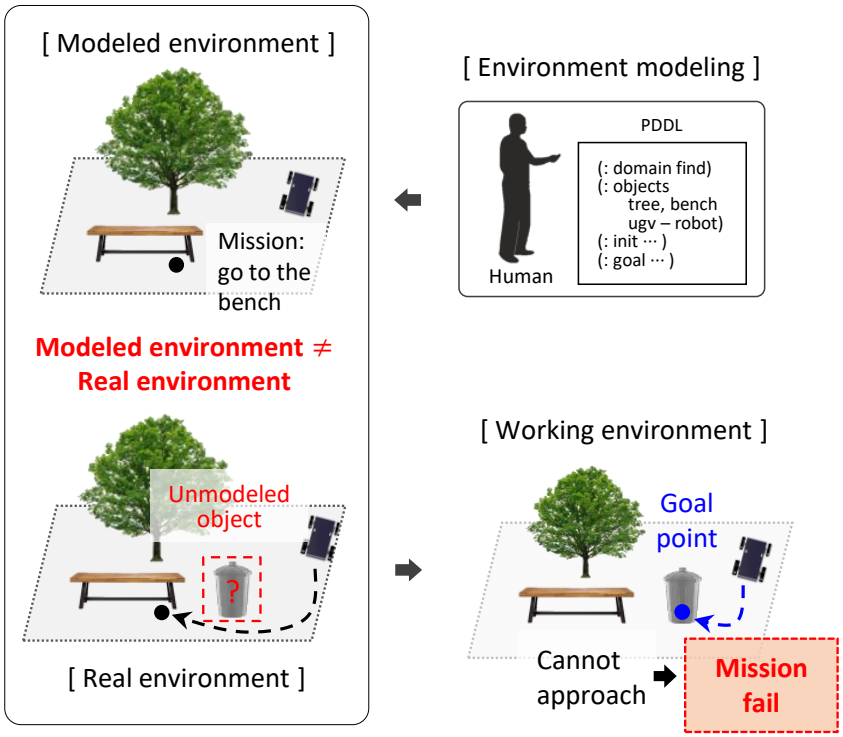


Figure 5.1: *Domain model acquisition problem:*

5.2 Related Work

Planning is an AI technology that seeks to organize activities in order to achieve specific goals [89]. It begins with a domain model describing the actions available to the planner in terms of their pre- and post-conditions, and a description of the current state is also provided. Given its goal, it constructs a plan by organizing instances of

actions into a structure that is causally valid and is predicted to achieve the goal while optimizing a cost function.

The family of Planning Domain Definition Languages (PDDLs) was defined to provide flexible languages for specifications of domain models and problem instances. Starting with problems described in STRIPS notations, where only predicates can be used to define the system states, PDDLs have grown to provide enormous expressive power, including large fragments of first-order logic to combine propositional expressions and numerical state variables to support the handling of real-valued quantities, and constraints to impose additional conditions on sets of valid plans. For instance, metric planning introduces planning with costs, while temporal planning covers the action duration (version PDDL2.1). PDDL3.0 allows planning with preferences and soft goals. Finally, processes and events enable mixed discrete–continuous behavior and complex dynamics to be captured through the latest extension of the language (PDDL+).

5.3 PDDL Planning with Incomplete World Knowledge

The overview of the proposed autonomous planning method for incomplete world knowledge is illustrated in Figure 5.2. Human experience-based corpus information composed in the form of natural language is transformed into the form of the resource description framework (RDF) triple, which is a subject-predicate-object structure. Then, words are encoded into the vector space to find similarities among them. In this dissertation, I use a method that encodes the data using not only the shapes of the characters, but also the meanings of the words. As a result, the relations between the words are generated and stored in the triple repository. The generated taxonomy is utilized in symbolic planning. If the information is insufficient for robots to achieve the goal, humans provide the missing information in the form of natural language. The provided information is added to the RDF graph as new information and used for mis-

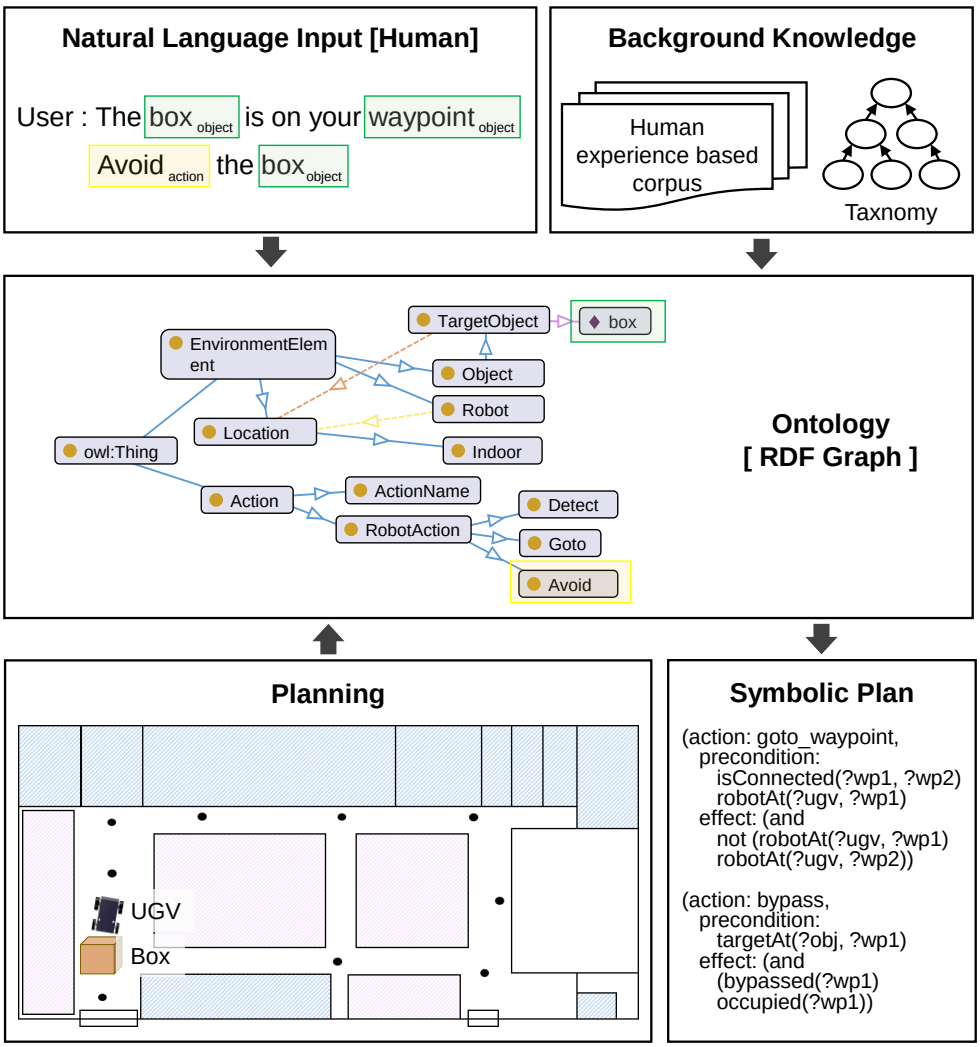


Figure 5.2: *Overview of autonomous planning with incomplete world knowledge*

sion planning. The information given by humans can facilitate high flexibility in the mission planning.

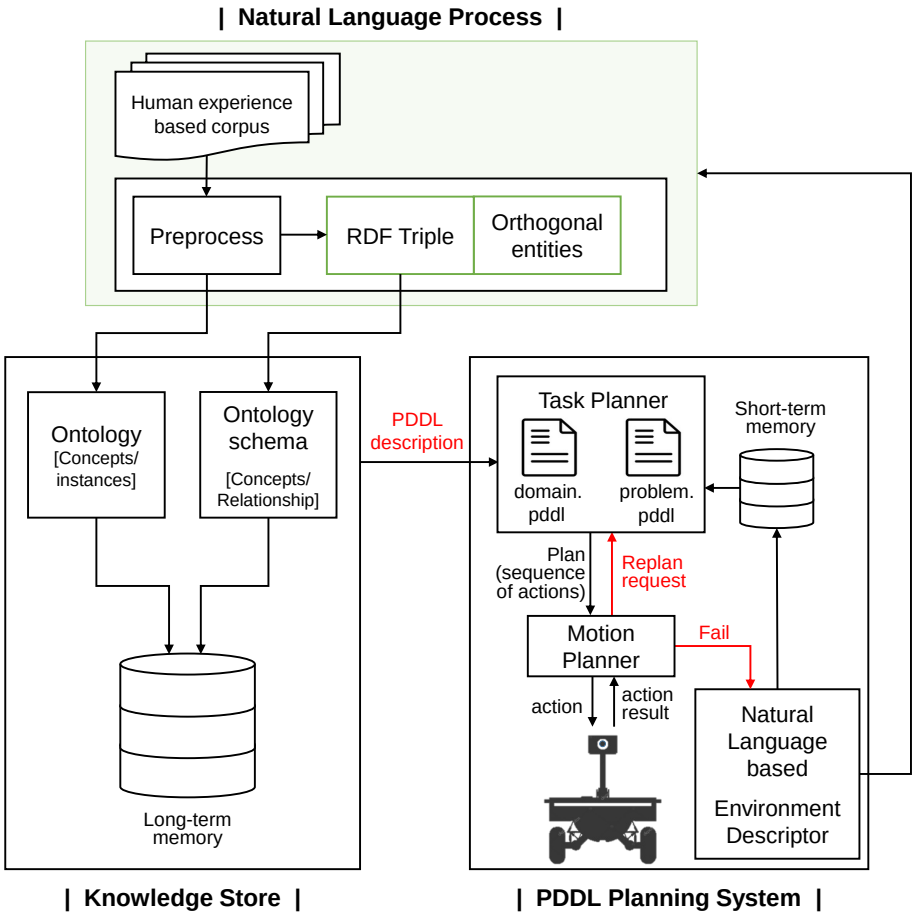


Figure 5.3: *Detailed architecture of the proposed method*

5.3.1 Natural Language Process for PDDL Planning

The detailed architecture of the proposed method is shown in Figure 5.3. I utilize the word synset dataset [90] and wikiHow dataset [91] for mission planning ontology construction. The word synset dataset provides a set of synonyms in the "synset." The wikiHow dataset lists the step-by-step actions to perform the tasks. OpenIE [92], which

is a natural language process tool, is used to express the natural language sentences in the form of subjects, verbs, and objects. These RDF triples are mapped into the vector spaces to be added to the ontology graph according to similarity and relationship. Many algorithms have been developed to map words into numerical vector spaces. The most basic mapping method is one-hot encoding that allows categorical data to be represented as numeric data. However, it is not an efficient way for the proposed method to represent natural language because the dimension of the vector increases as the number of words grows. The vector generated by one-hot encoding does not contain the meanings of the relationships between the words. However, for the purpose of this research, I require a learning method that efficiently maps words to low-dimensional vectors while considering the semantic meanings of words. Therefore, I utilize a mapping algorithm based on distributional semantics [87] to generate vectors considering relations between the words, and store the information in the ontology graph. PDDL domain and problem files are generated with the fetched information using SPARQL according to the goal mission constraints from the stored data. Then, the sequence of actions for robots is generated by the planner. The motion planner executes the actions and provides continuous feedback to the planner. If a robot fails in the mission because of environmental changes, it can achieve the mission goal by performing the overall process.

5.3.2 PDDL Planning System

In this dissertation, I plan to use PDDL2.1 [93] based on ROSPlan[88]. The ROSPlan framework is a highly modular set of tools used to embed Planning in the Robot Operating System (ROS) [94]. ROSPlan's objective is to link standard planning languages with ROS and to provide a modular framework in which different temporal planners can be easily used. For example, POPF, a forwards-chaining temporal planner [4], can be replaced by Temporal Fast Downward [95], LPG [96], UPMurphi [97], or any other planner capable of reasoning with the PDDL2.1 formalism. Non-temporal probabilis-

tic planners and planning models, among others, can also be exploited. For example, ROSPlan can be used with PDDL+, an extension of PDDL that describes exogenous events and continuous processes, Probabilistic-PDDL used in non-deterministic domains, and PDDL extended with sensing actions for use with conditional planning. Using ROSPlan, it is possible to automatically generate an initial state from the knowledge parsed from sensor data and stored in a knowledge management system, automate calls to a planner, and then post-process and validate the plan. Plan execution can be handled taking into account a changing environment and action failure. Planned actions can be matched to ROS action messages for lower level controllers. The configuration of the planning system is shown in Figure 5.4.

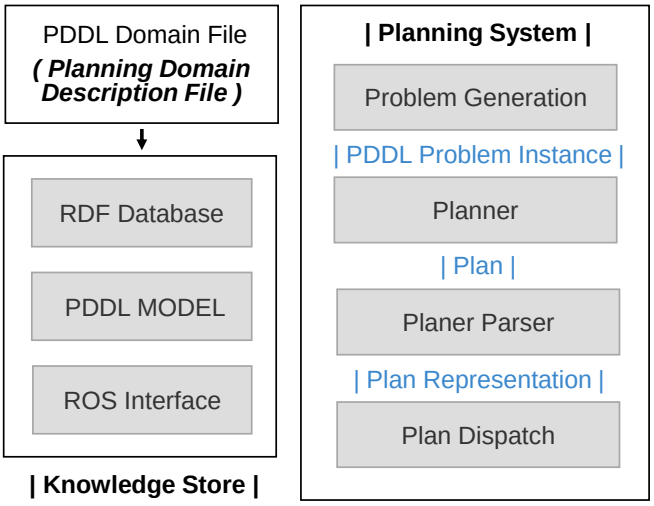


Figure 5.4: *Overview of the planning system*

5.4 Experiments

The size of the simulation environment is 110 m × 100 m, and includes indoor and outdoor areas. An unmanned ground vehicle (UGV) with a 2D laser sensor and an RGB-D camera are used for mission planning. The 2D laser sensor is used for simul-

taneous localization and mapping, and the RGB-D camera is used for semantic scene understanding (such as object recognition). The platform runs on Ubuntu16.04, ROS Kinetic, and Gazebo. Jena TDB, provided by Apache, is used to store the semantic memory, whereas MongoDB is used to store the robot's working memory. Details of software architecture for the proposed method are illustrated in Figure 5.5.

Experiments are conducted on two known scenarios, partially known workspace and acquiring missing knowledge. I evaluate the effectiveness of the proposed method by validating the generated plan's executability based on natural language sentences provided by humans. In the scenario where missing knowledge is acquired, the proposed method is verified by whether it includes unmodelled concepts in the ontology graph and whether the robot achieves the goal. I consider the mission failure case because of insufficient modelled information. For example, I assume that the robot's situation is such that only a few concepts of the surrounding environment are modelled and it cannot achieve the goal with this information. The necessary information is provided by humans in natural language. The provided data are expressed in triple form and added to the ontology graph to generate PDDL files to enable the robot to finish the mission successfully.

The partially known workspace scenario is used to verify whether new data are properly added to the existing modelled domain. For example, suppose a robot performs the mission "Search all unoccupied points in the indoor area" within a working area that is modelled differently from the actual environment. The robot will fail to achieve its goal. Using the proposed algorithm, the robot can obtain the necessary information from humans to accomplish the mission. The initial ontology graph is illustrated in Figure 5.6. The green box in Figure 5.7 shows the result of a partially known workspace scenario derived from "avoid the object," while the red box in Figure 5.7 shows the result of acquiring missing knowledge in the scenario "A box is on the waypoint."

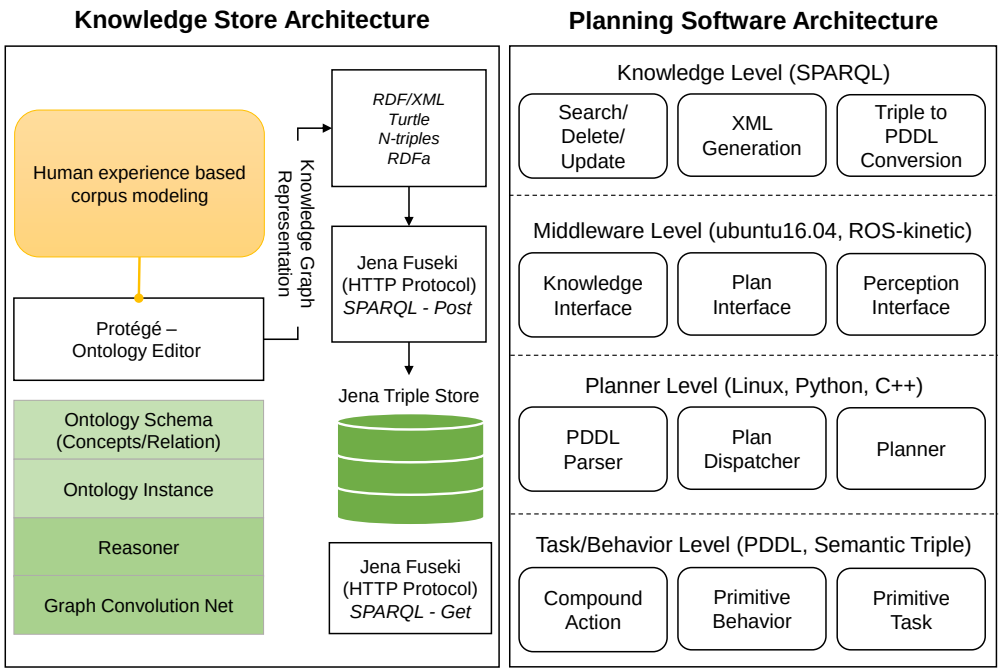


Figure 5.5: *Software architecture for the proposed method*

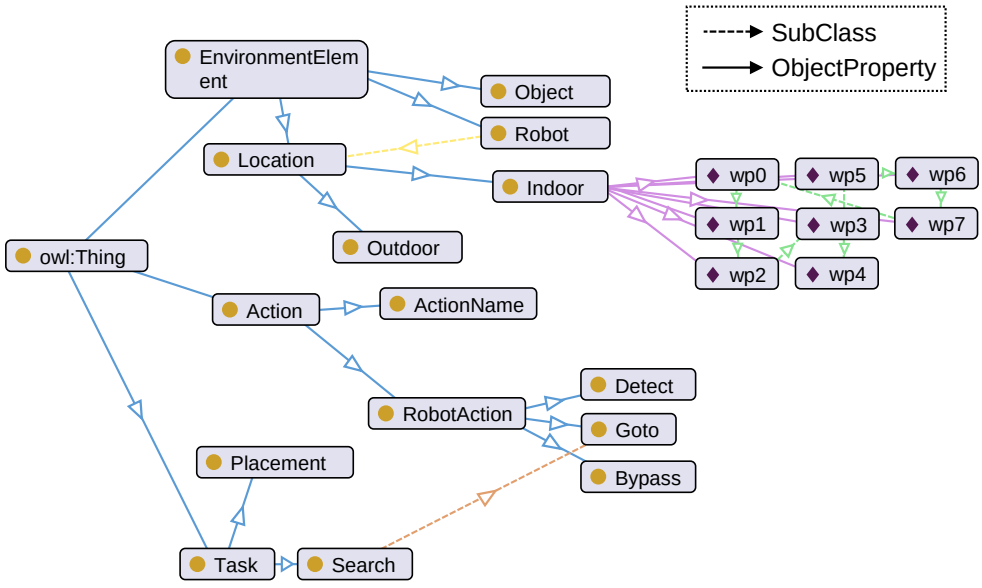


Figure 5.6: *Initial ontology graph*

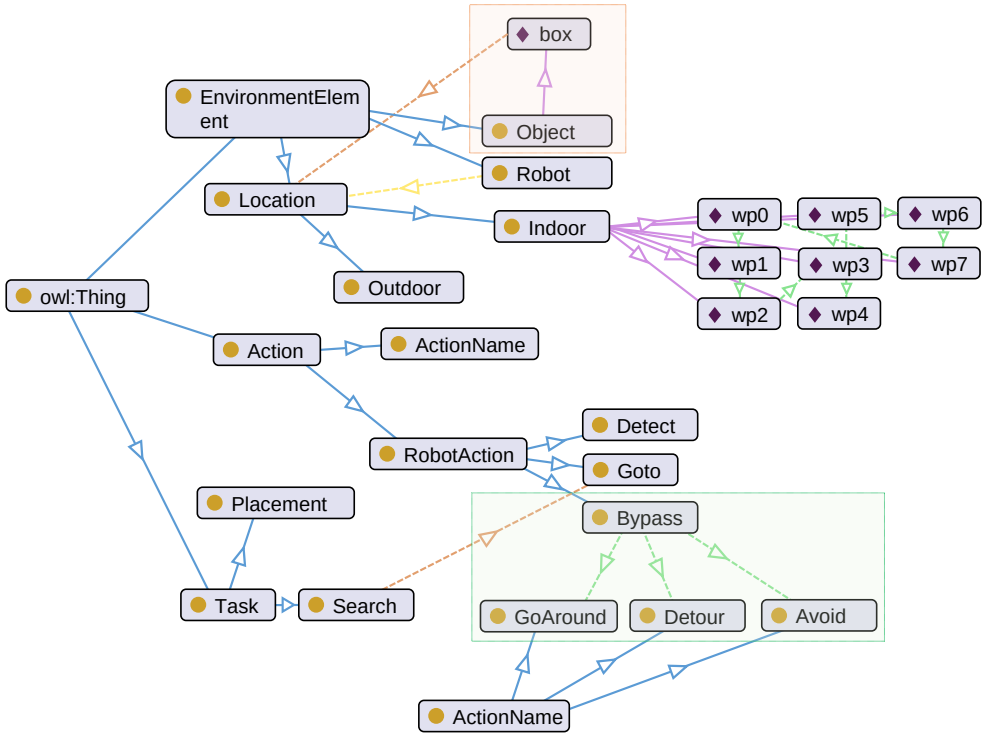


Figure 5.7: *Generated ontology graph based on the proposed method*

5.5 Summary

This dissertation proposes a method for robots to automatically create a series of actions using natural language, namely unstructured data, provided by humans based on ontology graphs. Dynamically changing environments can cause robots to fail at mission planning occasionally. Information provided by humans can increase the success rate of mission planning, thereby achieving the goal. Unstructured sentences are expressed as RDF triples by utilizing natural language grounding technology. Then, the generated triple data are added to the existing ontology graph. The robot generates the PDDL domain and PDDL problem file using the added information, and completes the mission by executing the sequence of actions generated by the planner. Thus, the results of my dissertation show that the proposed human–robot cooperation method provides flexibility in robot mission planning.

Chapter 6

PDDL Planning with Natural Language-Based Scene Understanding

Natural-language-based scene understanding can enable heterogeneous robots to cooperate efficiently in large and unconstructed environments. However, studies on symbolic planning rarely consider the semantic knowledge acquisition problem associated with surrounding environments. However, recent developments in deep learning show outstanding performance in semantic scene understanding using natural language. In this dissertation, a cooperation framework that connects deep learning techniques and a symbolic planner for heterogeneous robots are proposed. I employ neural networks for natural-language-based scene understanding to share environmental information among robots. I then generate a sequence of actions for each robot using planning domain definition language (PDDL) planner. JENA-TDB is used for data acquisition store. The proposed method is validated using the simulation results obtained with one unmanned aerial and three ground vehicles.

6.1 Introduction

Natural-language-based scene understanding is a critical issue for symbolic planning for heterogeneous multi-robot cooperation. I can mitigate the knowledge acquisition

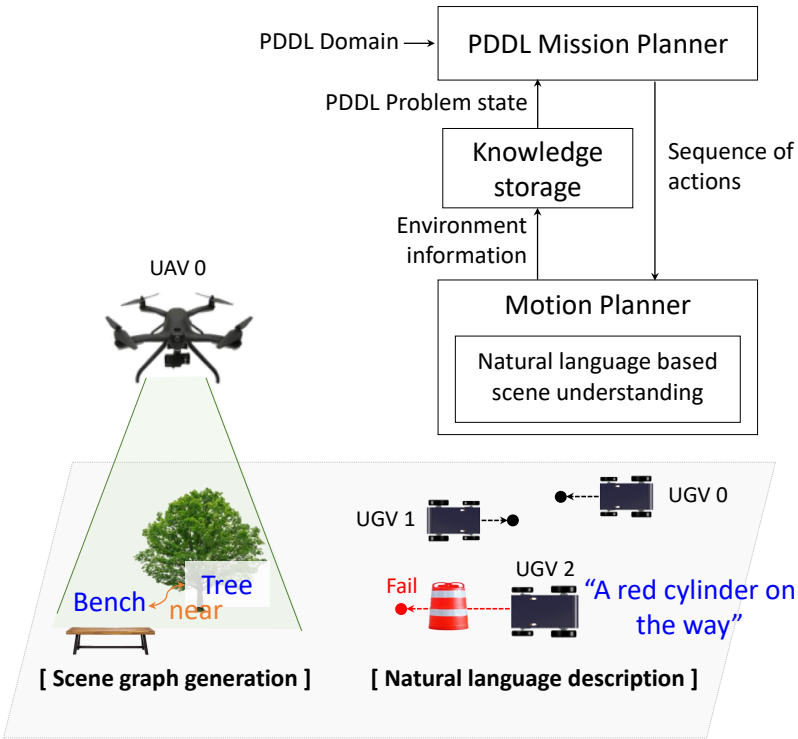


Figure 6.1: *Outline of my approach: PDDL mission planner is utilized to generate the sequence of actions using the PDDL domain and environment information for heterogeneous robots. The surrounding environment is represented as a scene graph. If a robot fails the mission, it generates a natural language description.*

problem associated with the area of symbolic planning by sharing the environmental information expressed in natural language with diverse robots. Recently, heterogeneous multi-robot systems composed of robots with different abilities have received increasing attention as they are required in a broad range of fields such as surveillance, environment exploration, and field robotics [98]. Various symbolic planning studies have been conducted to generate a sequence of actions for each robot to achieve shared missions. Miranda et al. [99] embedded a symbolic task planner using the PDDL in the robot operating system (ROS) for multi-robot navigation. Zhang et al. [100] presented a multi-robot symbolic planning system with an iterative inter-dependent algorithm to find optimal plans that minimize the overall cost. Compared to many studies that aimed to maximize overall utility and reduce costs during identification of optimal plans for multi-robots, the environmental information sharing method between robots can mitigate environmental knowledge acquisition problems but continues to be insufficiently studied. Corah et al. [3] employed a Gaussian mixture model to map the surrounding environment while maintaining a low volume of memory for communication-efficient planning. However, since this method uses an algorithm designed for a specific sensor, it poses a practical application issue for a heterogeneous multi-robot system composed of different processors, implementation techniques, and sensors. Moreover, since these methods share raw sensor data, the additional process needed to achieve meaningful information from the sensor data imparts inefficiency to the overall process. Therefore, in this dissertation, I propose a symbolic planning method that shares natural-language-based environment information containing semantic meaning, rather than raw sensor data, for heterogeneous multi-robot cooperation.

Semantic scene understanding in objects or natural languages, rather than in points, lines, and planes that cannot contain semantic meanings, is widely researched in the fields of robotics and computer vision [8, 101, 102]. Zhang et al. [9] generated semantic maps with object-level entities using the semantic simultaneous localization and mapping (SLAM) algorithm. Karpathy and Fei-Fei [46] generated dense captions for

multiple regions and the overall area in an image using bidirectional recurrent neural networks (RNNs) and a multimodal RNN. Yao et al. [103] found semantic and spatial relationships between objects in images through graph convolutional networks (GCNs) and long short-term memory (LSTM). The results of this semantic information are utilized for various applications such as robot navigation [11], image retrieval [104], and question and answering [105]. However, the application of heterogeneous multi-robot cooperation planning is not considered.

On the one hand, deep learning outperforms extraction of semantic information from an unseen environment, but it is difficult to learn high-level processes that require causal reasoning, analogical reasoning, or planning using data [106]. On the other hand, symbolic planning that uses a logic model can guarantee solution optimality, but it can only be applied to a predefined environment. To combine deep learning and classical planning, Asai and Fukunaga [107] encoded images as latent vectors with a variational autoencoder and applied PDDL planning. Mao et al. [108] proposed a neuro-symbolic concept learner that learns visual scenes using a neural network and expresses them in an executable form in symbolic programs. In this study, symbolic planning and deep learning techniques are integrated to propose a cooperation planning architecture with natural language scene understanding for a heterogeneous robot team, as shown in Figure 6.1. Convolutional neural networks (CNNs), GCNs, and RNNs are used for natural language description and scene graph generation. JENA-TDB is used to share the semantic representation of the environment among the robots. The planning phase of ROSPlan [88] is used for generating plans. The proposed method is verified by a simulation using one unmanned aerial vehicle (UAV) and three unmanned ground vehicles (UGVs).

6.2 Related Work

This dissertation is related to studies of heterogeneous multi-robot cooperation planning and natural-language-based semantic scene understanding, the idea being to connect symbolic planning and deep learning.

Heterogeneous multi-robot cooperation planning

The multi-robot system has the advantage that it can perform complex tasks that cannot be accomplished by one single powerful robot with many capabilities through cooperation [109]. For example, a large building can be cleaned with one robot, but it is time-consuming and unrealistic. Thus, a multi-robot system that dispatches the overall mission into smaller sub-problems to individual robots is necessary. [98] proposed a cooperative control scheme for a heterogeneous ground-air robot team. [110] integrated a temporal planning approach with a PDDL planner for heterogeneous teams of robots. [111] solved the decision-making issues of aerial robots using an integrated decision-making framework. [86] represented the environment in a scalar field and created a time-optimized mission plan for UGVs using a cascaded heuristic optimization algorithm. However, most studies of heterogeneous multi-robot systems focus on achieving shared goals effectively, with minimum time and cost, through algorithms rather than acquiring knowledge of the environment using multiple robots.

Some researchers tried to solve the environmental knowledge acquisition problem through data sharing between each robot. [112] used an adaptive transmission method for efficient distributed information sharing. [85] proposed a shared information integration method for cooperative environment data gathering. [113] introduced two approaches that can learn how information may be shared: reinforced inter-agent learning and differentiable inter-agent learning. These studies share raw sensor information that can hardly infer semantic meanings without algorithms. They need to be designed suitably for each robot in a heterogeneous multi-robot team. Unlike conven-

tional studies, I introduce a method that acquires environmental knowledge in the form of natural language and applies to multi-robot cooperation planning.

Natural language-based scene understanding

Many studies on robotics have proposed graph-based simultaneous localization and mapping (SLAM) using semantic scene understanding and various sensors. [114] performed object recognition using range data and feature-based semantic SLAM with a UAV. [115] proposed a dense 3D SLAM system composed of stereo-ORB-SLAM and a CNN for a traffic environment. [116] combined a matured SLAM system named RTAB-Map and a CNN to utilize depth image information. However, they rarely considered the natural language inference problem, which is important in multi-robot communication.

However, semantic scene understanding using natural languages such as image captioning, visual question and answering (VQA), and scene graph generation is widely studied in the field of computer vision. Lu et al. [117] generated image captions using an attention-based neural encoder-decoder framework. Lu et al. [118] utilized a co-attention model in a hierarchical fashion to perform VQA. Dai et al. [31] proposed a deep relational network that can exploit the statistical dependencies of detected objects and their relationships. Since these approaches use images as inputs, graph maps, which are widely used as environment representation by robots, are rarely utilized. This dissertation proposes an architecture that includes natural language description and scene graphs generated using a graph map in multi-robot planning.

Connecting symbolic planning and deep learning

Many studies of robotics involving mission planning with symbolic planners have been conducted. Srivastava et al. [119] demonstrated off-the-shelf task implementation with a PDDL planner. Dornhege et al. [120] applied geometric reasoning to symbolic planning and conducted real-world mobile manipulation experiments. Manso et al. [121]

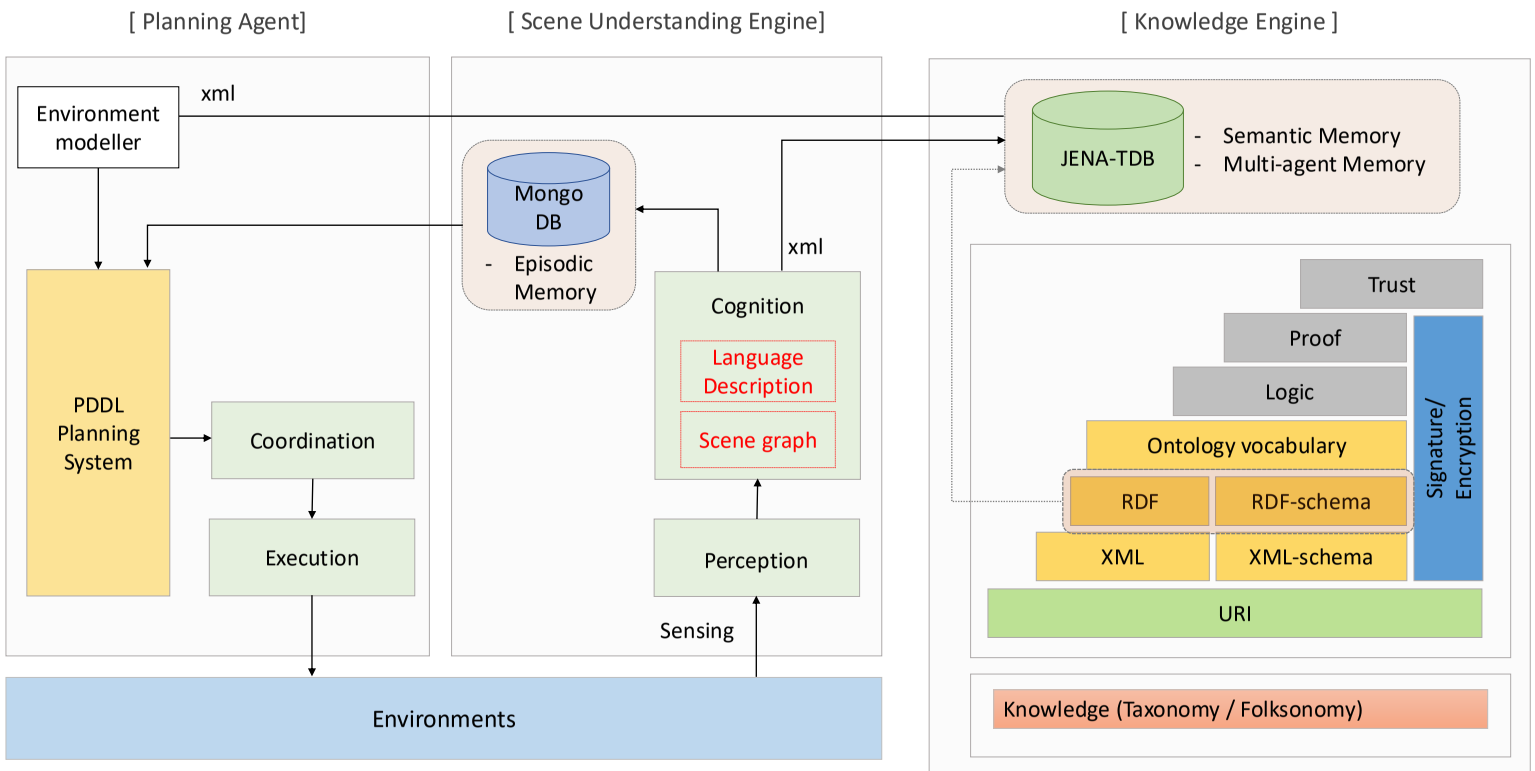


Figure 6.2: *General overview of the proposed architecture*

utilized graph models and graph rewriting rules with a symbolic planner for human–robot interaction. However, symbolic planning is hardly applied to new, unforeseen, and dynamic environments, because the environments should be modelled directly by a human or via a compiler. However, deep learning, which is a data-driven approach, has shown outstanding performance in environmental cognition [122, 123]. To take advantage of both fields, Zhang and Sornette [124] introduced a deep symbolic network to represent any knowledge as a symbol. Liao and Poggio [125] converted objects into symbols using an object-oriented deep learning algorithm. They focused on generating symbols using deep learning, rather than setting the overall architecture for planning. In this study, I propose a method to bridge the gap between symbolic planning and deep learning techniques, and verify it using heterogeneous multi-robot cooperation planning.

6.3 A Framework for Heterogeneous Multi-Agent Cooperation

This section explains the framework devised to connect deep learning techniques and the symbolic planner for cooperation among heterogeneous agents. Unlike conventional planning systems for robots [88], my framework entails natural-language-based cognition and a knowledge engine for multiple agents. The general overview of the framework is shown in Figure 6.2. It is composed of perception, cognition, planning, coordination, execution, and memory storage. Perceptively, sensor information obtained from environments is continuously passed to cognition. During cognition, scene understanding-based natural language is created by generating language description and scene understanding using deep learning techniques. Then, the generated semantic information is passed on to the knowledge engine while raw sensor data are sent to episodic memory storage. Using the episodic memory and knowledge collected from multiple robots, the PDDL planning agent builds a sequence of actions for each agent.

Then, the robots complete the required actions through coordination and execution. The details are as follows.

6.3.1 Natural Language-Based Cognition

To understand the surrounding environment in natural language, I generate a natural language description and scene graph. In this study, I assume that the robots use a graph map (for motion planning) generated using semantic SLAM, which is a widely used environment representation method in robotics [8]. To utilize the graph map $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that contains features and positions of the detected object as nodes $v_i \in \mathcal{V}$ and their relationships as edges $e_{ij} = (v_i, v_j) \in \mathcal{E}_{ij}$, I closely follow Moon and Lee [126] for generating the language description and graph inference phase of Xu et al. [34] for the scene graph generation. However, since the edge information of the graph map is binary, which can only infer whether a connection exists or not, or a weighted value that indicates relations such as the Euclidean distance between objects, it is difficult to find the semantic meaning. Therefore, I additionally extract features of the union region of two objects for edge information. For each v_i and e_{ij}, features are extracted using VGGNet [51]. f_i^v is the feature vector of v_i, and f_{ij}^e is the feature vector of e_{ij}. p_i is position vector of v_i.

The neural network architecture for language description is illustrated in Figure 6.3. First, a GCN with graph convolution layers defined by spectral graph theory and fully connected layers is utilized to extract features from irregular and non-Euclidean graphs. Then, an RNN is used to generate a language description over the graph. The RNN takes the encoded graph features concatenated with a word vector and predicts the probabilistic distribution of the target word vector. Given that I also back-propagate the GCN when training the RNN, I can expect that graph features that fit the generated sentence will be extracted. The generated description can be used to understand the surrounding environment when an unexpected situation occurs.

Scene graph generation involves the process of finding appropriate words corre-

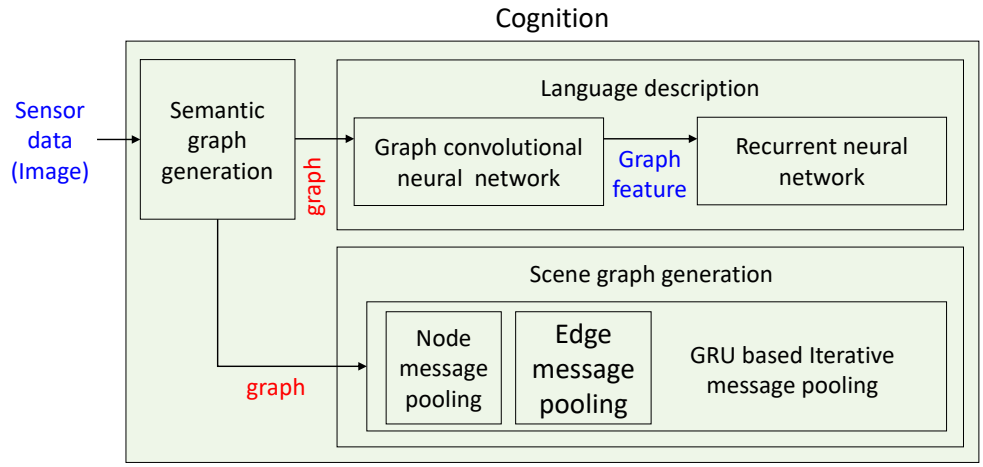


Figure 6.3: *Neural network architecture for language description: A GCN extracts features from the graph map. The extracted graph feature is concatenated with a word and feed into the RNN as input. Then, the RNN generates sentence attention over the graph.*

sponding to each node and edge of the graph. I denote variables that need to be predicted as $\mathbf{g} = (v_i^{class}, e_{ij} \mid i = 1 \ldots n, j = 1 \ldots n, i \neq j)$, where $\mathcal{C}$ is a set of object classes and $\mathcal{R}$ is a set of relationship types, $v_i^{class} \in \mathcal{C}, e_{ij} \in \mathcal{R}$. The optimal g^* is found as follows:

$$\mathbf{g}^* = argmax_{\mathbf{g}} Pr(\mathbf{g} \mid f_i^v, f_{ij}^e) \tag{6.1}$$

$$Pr(\mathbf{g} \mid f_i^v, f_{ij}^e) = \prod_{i \in V} \prod_{j \neq i} Pr(v_i^{class}, e_{ij} \mid f_i^v, f_{ij}^e) \tag{6.2}$$

The iterative message pooling method based on the gated recurrent unit (GRU) is utilized as shown in Figure 6.3. Edge features and node features are fed into the edge GRU and node GRU as the initial value, respectively. After the message pooling, the edge message is fed into the edge GRU and the node message is fed into the node GRU. The iteration that follows precisely predicts words for the nodes and edges. The scene graph can be used to gather environmental information in natural language for

large and unstructured environments.

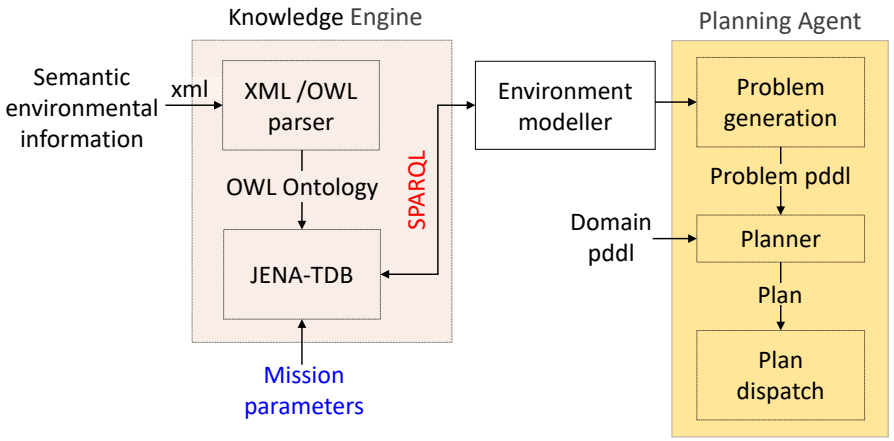


Figure 6.4: *Detailed architecture of the knowledge engine, environment modeler, and PDDL agent*

6.3.2 Knowledge Engine

The knowledge engine obtains semantic environmental information in XML and stores it in triple store, which uses a resource description framework (RDF) such as ”subject-predicate-object” or ”resource-property type-value” unlike the conventional relational database that saves data in ”key-value.” Triple store uses the SPARQL protocol and RDF query language (SPARQL) to create, read, update, and delete the graph data that contain relations between objects. The triple store facilitates the reasoning process by using the relations and attributes between objects to find new relations. In this study, I utilize JENA-TDB, a type of triple store. It is an open source framework developed by Apache for the manipulation of RDF data. JENA-TDB provides persistence storage for the RDF and web UI with the Apache Fuseki interface using the http protocol.

The XML/OWL parser in the knowledge engine parses the XML file into OWL Ontology. Ontology is a model that explicitly describes conceptual meanings by restricting the relations in the artificial intelligence field. OWL is one of the ontology

expression languages. It is designed to create an environment in which machines and agents can understand and utilize resources using reasoning and formal syntax. OWL defines the class and property of instances, describes relations between the classes and subclasses, and infers new concepts. In this study, I classify the topology and semantic relations among objects as object property relations and the attributes of the object as data property relations when the knowledge engine receives the XML file containing the taxonomy of classes and subclasses of semantic information achieved by cognition. The classified relations are described in OWL in the XML/OWL parser. The generated OWL ontology is saved in JENA-TDB using the Fuseki http protocol. When JENA-TDB receives a request from the environment modeler to hand over the required information to set the initial and goal states for mission planning, SPARQL is used to gather data. The proposed architecture of the knowledge engine, environment modeler, and PDDL agent is illustrated in Figure 6.4.

Table 6.1: Details of planning and replanning performed by the planning agent

```
1:  procedure PLAN DISPATCH (Domain D, Mission M)
2:    while M contains goals do
3:      I := generateProblem(D, M);
4:      P := plan(D, I)
5:      while execute do
6:        a := pop(P);
7:        dispatch(a)
8:      end while
9:      execute := execute ∧ actionSuccess(a)
10:   end while
11: end procedure
```

6.3.3 PDDL Planning Agent

I utilize the planning agent of [88] as the PDDL planning agent. ROSPlan provides planning in the robot operating system (ROS). However, since they hardly utilize nat-

ural language information achieved from surrounding environments, I modified it appropriate for my approach. In the planning agent, problem PDDL generation, plan generation, action dispatch, and replanning are performed. From the environment modeler and Mongo DB, data related to initial state and mission parameters are gathered and feed into problem generation. Then, the problem PDDL is automatically generated and handed to a planner with domain PDDL. In this dissertation, the POPF planner is used. Once the plan is generated, the plan dispatch parses the PDDL actions to the ROS messages for the robots to complete the overall plan. During the execution, if an action fails because of changes in the environment, the planning agent reformulates the problematic PDDL by replanning, as shown in Table 6.1.

6.4 Experiments

I demonstrate the proposed framework with a patrolling scenario and find the missing child using one UAV and three UGVs. The operational diagram for the proposed method is illustrated in Figure 6.5. It is composed of the control tower, natural language processor, simulator, and JENA-TDB. The scenario is run in the simulation to verify the proposed architecture. The details are as follows.

6.4.1 Experiment Setting

The simulation environment was designed as an area around REDONE technologies cooperation, as shown in Figure 6.6 and Figure 6.7 (a). The size of the area was $110m \times 100m$. I utilized three UGVs of REDONE technologies, each named Smart Cookie, and 1 UAV of REDONE technologies, named Beyond. Each Smart Cookie has 2D laser sensors and an RGB-D camera as illustrated in Figure 6.7 (b). Beyond is equipped with an RGB-D camera. The laser sensor is used for navigation on the execution part while the RGB-D cameras are used for cognition for the natural-language-based scene understanding. Each robot navigated using the generated map and sensor. The platform

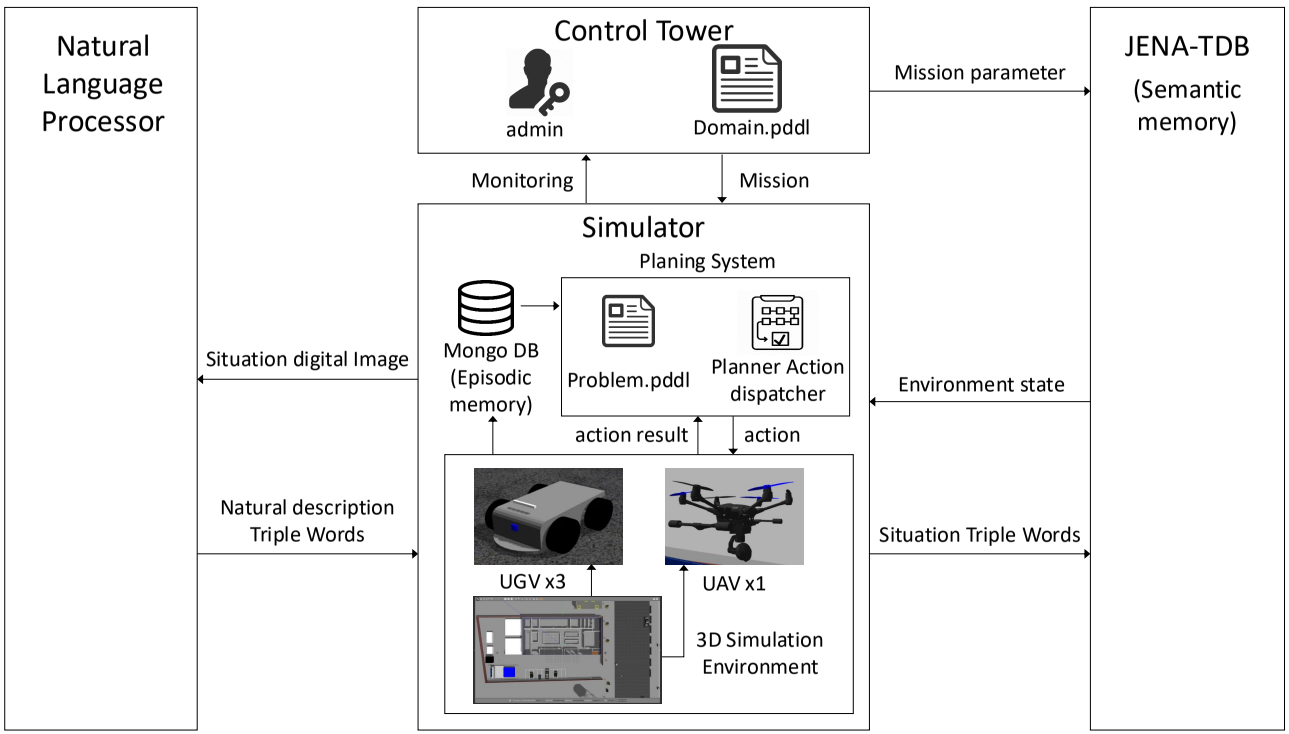


Figure 6.5: *Operational diagram for the proposed method: It consists of a control tower, natural language processor, simulator, and JENA-TDB. Planning and execution are performed by the simulator. Natural-language-based scene understanding is processed in the natural language processor. JENA-TDB is used as the semantic information processor.*

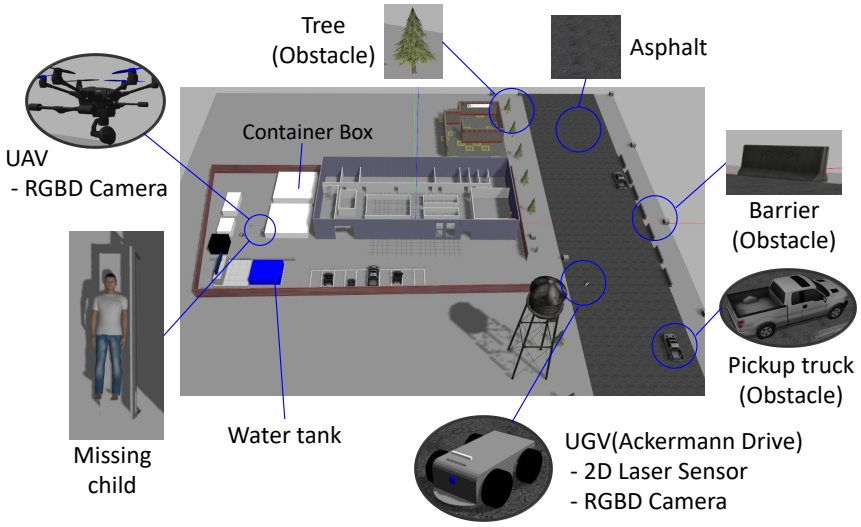
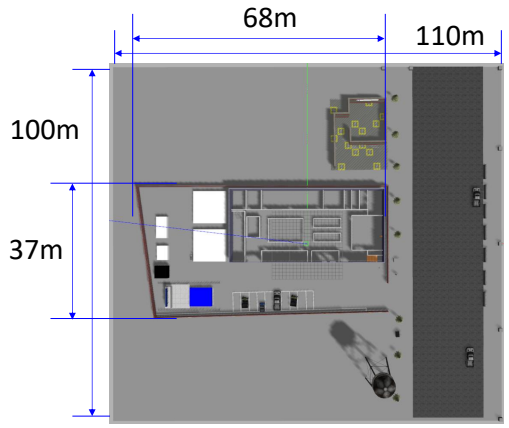


Figure 6.6: *Simulation environment*

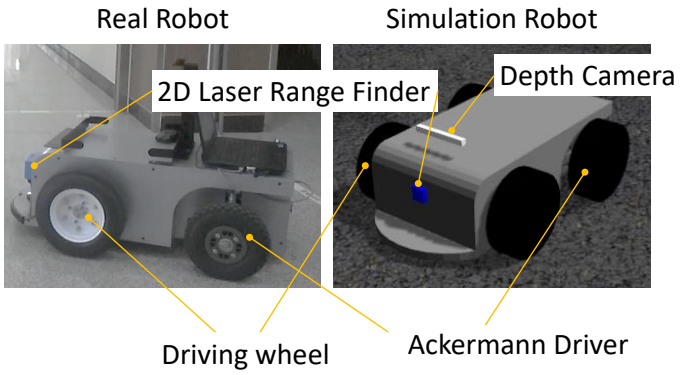
was set up with Ubuntu 16.04, ROS Kinetic, and Gazebo 7. JENA-TDB is used as the semantic memory and Mongo DB is used as the episodic memory. DICQ.R is the control tower. I used tensorflow library and Python for the natural language processing, whereas JAVA was used for JENA-TDB, and C++ was used for the simulator. Socket communication was utilized to transfer information between processors. To train the neural network for scene understanding, I used the COCO dataset and visual genome dataset for language description and scene graph generation, respectively. Since these datasets use images for natural language processing, I manually generated a graph using bounding boxes and train the networks.

6.4.2 Scenario

The overall scenario outline is illustrated in Figure 6.8. Two missions were performed. One involved patrolling the area, and the other was concerned with finding a missing child. While the robots were visiting the point of interest (POI) for patrolling, a mission to find a missing child was generated by the DICQ.R. Every robot was required to



(a)



(b)

Figure 6.7: *Details of simulation environment*

	Description	Activity
Name	Finding missing child	-
Preconditions	• UGV1,2,3, UAV at initial point	-
Flow of events	• Perform 'finding missing child' mission from control tower (A1)	1. Control tower command UGV 1,2,3, and UAV to find a missing child
		2. UGV 1,2,3, and UAV visit every POI to find a missing child
		3. A robot find an unmodeled object
		4. The robot generates a scene graph to add meaning to the unmodeled object (human) for planning
		5. The robot create POI at the position of human
		6. Control tower generate new mission for UGV1,2,3 to go to created POI to check that the found human is the missing child
		7. Mission completed
Expected situations	• A robot cannot approach POI due to an unmodeled obstacle (A2)	1. The robot generates natural language to describe the current situation to control tower
		2. Replanning
	• A robot found an object expected to be the missing child (A3)	1. Save the scene graph in JENA-TDB
		2. Using semantic information of JENA-TDB, generate new mission
Post-Condition	• Every robot send execution result to control tower	-

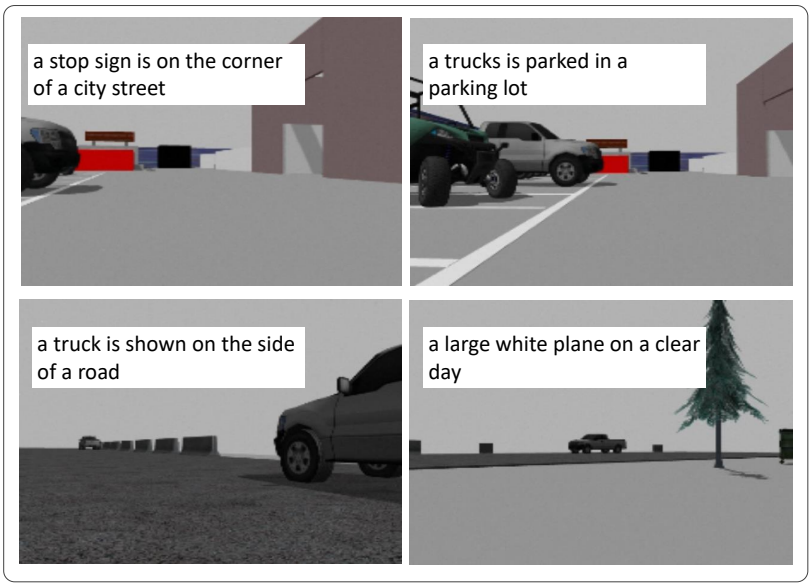
Figure 6.8: *Overall scenario outline: Using one UAV and three UGVs, patrolling was conducted and the missing child was found.*

report the current situation to the DICQ.R as well as if an unusual situation occurred. During the mission, I surmised what may happen if a dynamic obstacle, which a robot could not approach, were to suddenly appear at the POI. In this situation, the robot will generate natural language to report the current situation to the DICQ.R. Also, I expected at least one of the robots to find the missing child. In this case, I generated scene graphs to add POIs for the other robots to check. Analogously, the natural-language-based scene understanding can be applied to other planning missions.

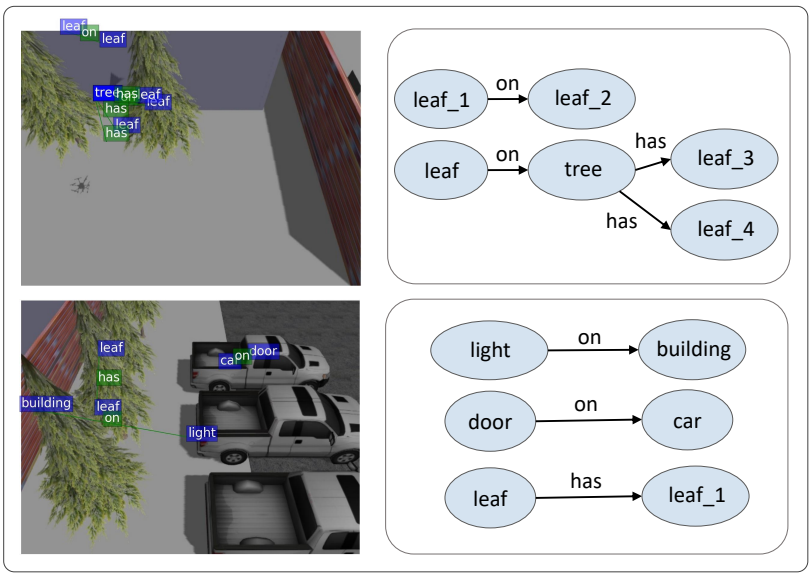
6.4.3 Results

The experiment involving patrolling and finding a missing child was successful. In this study, I used 16 POIs for robot patrolling according to the assigned area. When the child went missing, assume that a human is present at POI 9. Then, the robots were asked to check all the POIs and find a human who is likely to be the missing child. When such a human is detected, a scene graph is generated and sent to the DICQ.R. Using the achieved semantic information, a POI is added and the closest robot is asked to go to POI to check if the detected human is the missing child. Table 6.2 and Table 6.3 show the generated plans for the robots. In the initial plan, POI 16 is not included. After the human is detected by Beyond, a new POI (16) is generated and is checked by Smart cookie.

I used the XML file structure to send the semantic graphs to DICQ.R. The XML/OWL parser located inside the knowledge engine is provided triplet data that contain scene graph information. The OWL file is generated by the classification processes of object property and data property relations. The object property relation is relevant to the relationship between objects, and the data property relation is relevant to the properties of these objects. According to the command from the DICQ.R, which provides the mission parameters, JENA-TDB fetches semantic information using SPARQL and sends it to the environment modeler. For example, using the received triple data of ”human-behind-tree,” ”behind” is saved as ”owl:ObectProperty rdf:about plan:behind/.” ”human-



Exampls of generated language descriptions



Exampls of generated scene graphs

Figure 6.9: *Examples of generated language description and scene graphs*

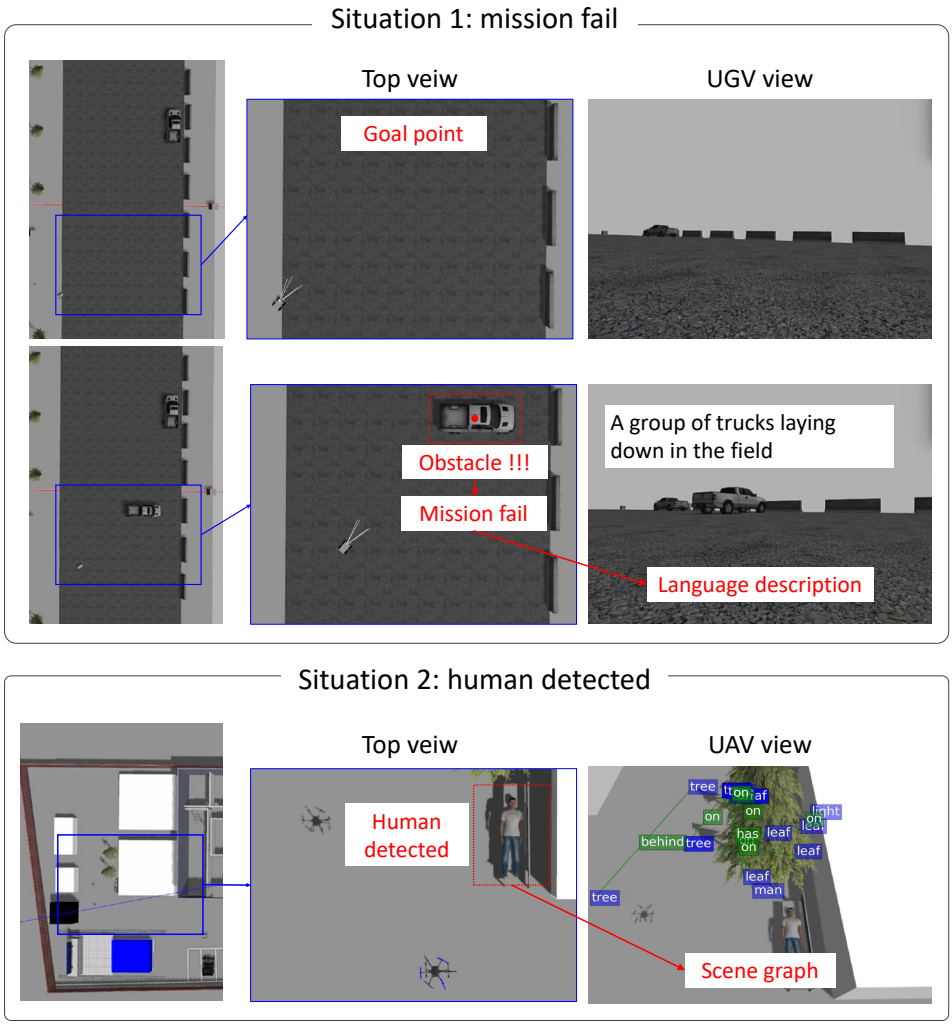


Figure 6.10: *Experiment results of the natural-language-based scene understanding across two situations: (1) language description in the "failed mission situation" to inform the control tower about the current situation, and (2) the scene graph in the "human detected situation,"*

Table 6.2: Generated plan for Part I of the scenario

0.000:	(goto_point_indoor cookie0 POI0 POI0)	[20.000]
0.000:	(goto_point_outdoor cookie1 POI12 POI12)	[20.000]
0.000:	(goto_point_street cookie2 POI2 POI2)	[20.000]
0.000:	(fly beyond0 POI6 POI6)	[20.000]
20.001:	(goto_point_indoor cookie0 POI0 POI1)	[20.000]
20.001:	(goto_point_outdoor cookie1 POI12 POI13)	[20.000]
20.001:	(goto_point_street cookie2 POI2 POI3)	[20.000]
20.001:	(fly beyond0 POI6 POI7)	[20.000]
40.001:	(goto_point_indoor cookie0 POI1 POI10)	[20.000]
40.001:	(goto_point_outdoor cookie1 POI13 POI14)	[20.000]
40.001:	(goto_point_street cookie2 POI3 POI4)	[20.000]
40.001:	(fly beyond0 POI7 POI8)	[20.000]
60.001:	(goto_point_indoor cookie0 POI10 POI11)	[20.000]
60.001:	(goto_point_outdoor cookie1 POI14 POI15)	[20.000]
60.001:	(goto_point_street cookie2 POI4 POI5)	[20.000]
60.001:	(fly beyond0 POI8 POI9)	[20.000]
80.001:	(detect beyond0 POI9 object)	[20.000]

Table 6.3: Generated plan for Part II of the scenario

0.000:	(goto_point_outdoor cookie1 POI15 POI16)	[20.000]

hasPositionX-100" is saved as "owl:DataProperty rdf:about plan:hasPositionX." Also, objects are parsed as "owl:NamedIndividual," which is used to describe instances.

The generated language descriptions and scene graphs are shown in Figs. 6.9. The language descriptions and scene graphs were successfully generated in the simulation environment. As illustrated in Figure 6.10, I utilized the natural-language-based scene understanding across two situations: (1) language description in the "failed mission situation" to inform the control tower about the current situation, and (2) the scene graph in the "human detected situation," to add a POI to verify whether the detected human is the missing child. As a result, I verified that the proposed framework could successfully perform the required planning using heterogeneous multiple robots based on natural-language-based scene understanding.

6.5 Summary

I proposed a new framework for heterogeneous multi-robot cooperation based on natural-language-based scene understanding. While other studies only used the raw sensor data for the purposes of perception, I focused on identifying semantic meanings from the surrounding environment to efficiently share information between heterogeneous agents. The framework combines deep learning and symbolic planning. Neural networks were used for the generation of semantic graphs and language descriptions. JENA-TDB was utilized to store semantic triple data. By gathering the data appropriate for mission parameters from JENA-TDB, the PDDL planner generated the sequence of actions for each robot. Using one UAV and three UGVs, the proposed method was successfully verified via simulation involving patrolling and finding a missing child.

Chapter 7

Conclusion

Semantic scene understanding is the process of perceiving environmental information in natural language or in a form that can provide semantic meanings. This is essential for the cooperation between humans and heterogeneous robots in that humans and robots can share information in an interpretable form to achieve a shared goal. In robotics, semantic mapping algorithms that generate graphs denoting features and positions of detected objects as nodes, have been widely studied recently. The graphs generated by these algorithms are unlike the maps generated by conventional methods, which comprise points, corners, lines, and planes. However, the generated semantic graphs do not find extensive application in data sharing methods for humans and robots. Therefore, there is a need for these graphs to be expressed in natural language. Natural-language-based scene understanding is investigated in various forms such as image captioning, visual question answering, and scene graph generation in the field of computer vision. However, these methods have not been widely applied to semantic graph maps used by robots to represent the surrounding environment. Moreover, they do not address the challenge associated with mission planning where humans and heterogeneous robots cooperate to achieve a common goal.

Humans and robots can improve work efficiency by sharing information of the surrounding environment in natural language. For example, let us assume that a robot

needs to move to another room and place a cup on a desk. However, if the cup is located on a chair, the robot fails to perform this task owing to perceptual error. However, if the robot can describe the environment and share it with the human, the success rate of the mission can be drastically increased. Moreover, increased network bandwidth and decreased QoS reliability because of robots sharing raw sensor data in a large environment can be addressed by sharing information in natural language, which is a compact and encoded semantic representation of the surrounding environment. In this dissertation, a scene graph, language description, and natural question were generated using a semantic graph map, and the obtained results of natural language processing were utilized by the system comprising human and heterogeneous robot.

Although semantic graph maps are commonly utilized in the investigation of the perceptual aspects of the environment, such maps are not extensively applied to natural language processing and question generation. Several studies have been conducted on the understanding of workspace images in the field of computer vision; these studies have automatically generated sentences; however, multiple scenes and 3D information were not utilized in these studies. Experiments were performed using publicly available datasets and the results indicated the superior performance of the proposed method.

Task planning for robots involves the use of incomplete and unreliable data. Observations made by sensors are used to update the model for task planning and execution through state estimation. An up-to-date model reduces the risk of plan failure and can promptly identify when a plan under execution is no longer valid. However, plans may fail during execution, and in such cases, it is critical that the robotic agent is able to exactly explain the reason for such failures.

The work presented in this dissertation can be practically integrated with task planning in two main ways. First, the generated scene graph can be used to update the model with new objects and relationships. Relationships in the scene graph can be used to update the (spatial) predicates that describe the current state in the planner's

model. Second, verbalization of the scene graph lead to enhanced descriptions of the state that can be used to describe the reason for plan failure. If a location is unreachable because of an obstruction, a verbalization of the scene graph can be provided to an operator as an explanation of the plan failure. This allows the operator to understand the ways in which the environment is different from the one expected, and what needs to be done next. In this section, we discuss future work in this direction.

The scene graph can be used to perform continuous updates to the current state through an integration with the planning sensor interface. This can automatically connect the scene graph generation of relations such as *light on building* into the predicates of the planning model. This integration has two main advantages: first, the spatial relations in the planner's model are kept up to date, which is necessary if the robot operates within a dynamic environment. Second, the newly detected objects can be immediately described in terms of their position and relationship with other objects. This is necessary for the planner to understand the ways in which the newly detected objects can be used in a plan, or the effect these objects might have on the state.

Plan execution performed by robots will be extended to include verbalization describing the plan under execution. This will be done by integrating the verbalization component with the execution components of the planning system in the following two ways: first, by providing verbalization of updates to the current state, and second, by providing verbalization of obstructions that prevent the robot from achieving its goal. In a human-robot system, it is critical that the human operator is given sufficient situational awareness to determine the state of the plan. By verbalizing the updates to the planner's model, an operator does not have to be an expert in the language of the domain model to understand what the robot is sensing. In addition, by verbalizing the reason for plan failure, the operator can quickly identify the unexpected event or object that resulted in the plan failure.

The work presented in this dissertation proposed a framework for human and heterogeneous cooperation based on natural language processing. In this framework, I

utilized the advantage of both symbolic planning, which can generate complete sequences of actions for multiple robots, and deep learning techniques, which exhibits outstanding performance in semantic scene understanding. The proposed method was verified through simulation with three UGVs and one UAV in a scenario involving patrolling and finding a missing child.

Bibliography

[1] N. Wang, Y. Zeng, and J. Geng, "A brief review on safety strategies of physical human-robot interaction," in *Proceedings of the ITM Web of Conferences*, vol. 25. EDP Sciences, 2019, pp. 1–3.

[2] S. Tellex, R. A. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics." in *Proceedings of the Robotics: Science and systems*, vol. 2, no. 3, 2014.

[3] M. Corah, C. Omeadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.

[4] A. J. Coles, A. I. Coles, M. Fox, and D. Long, "Forward-chaining partial-order planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2010.

[5] D. S. Nau, "Current trends in automated planning," *AI magazine*, vol. 28, no. 4, pp. 43–43, 2007.

[6] S. Miglani, "Nl to pddl: One-shot learning of planning domains from natural language process manuals," 2019.

[7] R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," *Artificial intelligence*, vol. 2, no. 3-4, pp. 189–208, 1971.

[8] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 1722–1729.

[9] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic slam based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75 545–75 559, 2018.

[10] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, "Semantic monocular slam for highly dynamic environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2018, pp. 393–400.

[11] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "A framework for learning semantic maps from grounded natural language descriptions," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1167–1190, 2014.

[12] Z. Hu, J. Pan, T. Fan, R. Yang, and D. Manocha, "Safe navigation with human instructions in complex scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 753–760, 2019.

[13] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 2048–2057.

[14] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, "Semantic compositional networks for visual captioning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5630–5639.

[15] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel, "Image captioning and visual question answering based on attributes and external knowledge,"

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1367–1381, 2017.

[16] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the International Conference on Computer Vision*, 2017.

[17] M. Klawonn and E. Heim, "Generating triples with adversarial networks for scene graph construction," in *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, 2018.

[18] A. Sato, J. K. Tan, and Y. Ono, "Development of a human-robot cooperative system based on visual information," in *Proceedings of the International Workshop on Advanced Image Technology*, vol. 11049. International Society for Optics and Photonics, 2019, p. 1104940.

[19] Q. Liu, Z. Liu, W. Xu, Q. Tang, Z. Zhou, and D. T. Pham, "Human-robot collaboration in disassembly for sustainable manufacturing," *International Journal of Production Research*, pp. 1–18, 2019.

[20] N. Nikolakis, V. Maratos, and S. Makris, "A cyber physical system (cps) approach for safe human-robot collaboration in a shared workplace," *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 233–243, 2019.

[21] S. M. Lukin, F. Gervits, C. J. Hayes, A. Leuski, P. Moolchandani, J. G. Rogers III, C. S. Amaro, M. Marge, C. R. Voss, and D. Traum, "Scoutbot: A dialogue system for collaborative navigation," *arXiv preprint arXiv:1807.08074*, 2018.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[23] X. Wang, X. Hua, F. Xiao, Y. Li, X. Hu, and P. Sun, "Multi-object detection in traffic scenes based on improved ssd," *Electronics*, vol. 7, no. 11, p. 302, 2018.

[24] X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan, "Dense semantic embedding network for image captioning," *Pattern Recognition*, vol. 90, pp. 285–296, 2019.

[25] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4894–4902.

[26] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6077–6086.

[27] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 670–685.

[28] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: An efficient subgraph-based framework for scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 335–351.

[29] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1745–1752.

[30] V. Ramanathan, C. Li, J. Deng, W. Han, Z. Li, K. Gu, Y. Song, S. Bengio, C. Rosenberg, and L. Fei-Fei, "Learning semantic relationships for better action retrieval in images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1100–1109.

[31] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3076–3086.

[32] Y. Li, W. Ouyang, X. Wang, and X. Tang, "Vip-cnn: Visual phrase guided convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 7244–7253.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[34] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5410–5419.

[35] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[36] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "Ikeabot: An autonomous multi-robot coordinated furniture assembly system," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 855–862.

[37] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*. IEEE, 2010, pp. 251–258.

[38] D. L. Chen and R. J. Mooney, "Learning to interpret natural language navigation instructions from observations." in *Proceedings of the Association for the Advancement of Artificial Intelligence*, vol. 2, 2011, pp. 1–2.

[39] S. Hemachandra, M. R. Walter, S. Tellex, and S. Teller, "Learning spatial-semantic representations from natural language descriptions and scene classifications," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 2623–2630.

[40] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multi-layer neural networks for object recognition," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 329–344.

[41] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2278–2283.

[42] A. Mallya and S. Lazebnik, "Recurrent models for situation recognition," *arXiv preprint arXiv:1703.06233*, 2017.

[43] X. Chen and C. Lawrence Zitnick, "Mind's eye: A recurrent visual representation for image caption generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2422–2431.

[44] D. Lin, C. Kong, S. Fidler, and R. Urtasun, "Generating multi-sentence lingual descriptions of indoor scenes," *arXiv preprint arXiv:1503.00064*, 2015.

[45] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 3156–3164.

[46] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[47] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang, "Ppr-fcn: weakly supervised visual relation detection via parallel pairwise r-fcn," *arXiv preprint arXiv:1708.01956*, 2017.

[48] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation recognition with graph neural networks," *arXiv preprint arXiv:1708.04320*, 2017.

[49] L. Juan and G. Oubong, "Surf applied in panorama image stitching," in *Proceedings of the 2nd International Conference on Image Processing Theory Tools and Applications*. IEEE, 2010, pp. 495–499.

[50] B. Leng, S. Guo, X. Zhang, and Z. Xiong, "3d object retrieval with stacked local convolutional autoencoder," *Signal Processing*, vol. 112, pp. 119–128, 2015.

[51] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[52] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[53] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[54] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[55] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2695–2702.

[56] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[57] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[58] M. Cakmak and A. L. Thomaz, "Designing robot learners that ask good questions," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2012, pp. 17–24.

[59] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4089–4098.

[60] N. Duan, D. Tang, P. Chen, and M. Zhou, "Question generation for question answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 866–874.

[61] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4651–4659.

[62] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler, "Learning to act properly: Predicting and explaining affordances from images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 975–983.

[63] J. Thomason, J. Sinapov, M. Svetlik, P. Stone, and R. J. Mooney, "Learning multi-modal grounded linguistic semantics by playing" i spy"." in *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2016, pp. 3477–3483.

[64] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699–704, 2003.

[65] S. Rosenthal, A. K. Dey, and M. Veloso, "How robots' questions affect the accuracy of the human responses," in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2009, pp. 1137–1142.

[66] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[67] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 974–983.

[68] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steady-states of iterative algorithms over graphs," in *Proceedings of the International Conference on Machine Learning*, 2018, pp. 1114–1122.

[69] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proceedings of the International Conference on Learning Representations*, 2018.

[70] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1225–1234.

[71] T. Ma, J. Chen, and C. Xiao, "Constrained generation of semantically valid graphs via regularizing variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 7113–7124.

[72] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[73] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, "Towards ai-complete question answering: A set of prerequisite toy tasks," *arXiv preprint arXiv:1502.05698*, 2015.

[74] Y. Zhang and M. Rabbat, "A graph-cnn for 3d point cloud classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 6279–6283.

[75] A. Gupta and P. Mannem, "From image annotation to image description," in *Proceedings of the International Conference on Neural Information Processing*. Springer, 2012, pp. 196–204.

[76] S. Zhang, L. Qu, S. You, Z. Yang, and J. Zhang, "Automatic generation of grounded visual questions," *arXiv preprint arXiv:1612.06530*, 2016.

[77] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015, pp. 2953–2961.

[78] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 1682–1690.

[79] K. Uehara, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada, "Visual question generation for class acquisition of unknown objects," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 481–496.

[80] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende, "Generating natural questions about an image," *arXiv preprint arXiv:1603.06059*, 2016.

[81] C. Nieto-Granda, J. G. Rogers, A. J. Trevor, and H. I. Christensen, "Semantic map partitioning in indoor environments using regional analysis," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1451–1456.

[82] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, 2007.

[83] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[84] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding," *ArXiv e-prints*, Feb. 2017.

[85] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Advances in Neural Information Processing Systems*, 2018, pp. 7254–7264.

[86] N. Kingry, Y.-C. Liu, M. Martinez, B. Simon, Y. Bang, and R. Dai, "Mission planning for a multi-robot team with a solar-powered charging station," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5233–5238.

[87] A. Lindsay, J. Read, J. F. Ferreira, T. Hayton, J. Porteous, and P. Gregory, "Framer: Planning models from natural language action descriptions," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2017.

[88] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, and M. Carreras, "Rosplan: Planning in the robot op-

erating system," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2015.

[89] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[90] J. Tomason and R. J. Mooney, "Multi-modal word synset induction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.

[91] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," *arXiv preprint arXiv:1810.09305*, 2018.

[92] G. Stanovsky, I. Dagan, *et al.*, "Open ie as an intermediate structure for semantic tasks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015, pp. 303–308.

[93] M. Fox and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.

[94] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *Proceedings of the International Conference on Robotics and Automation workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[95] P. Eyerich, R. Mattmüller, and G. Röger, "Using the context-enhanced additive heuristic for temporal and numeric planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2009.

[96] A. Gerevini and I. Serina, "Lpg: A planner based on local search for planning graphs with action costs." in *Proceedings of the International Conference on Artificial Intelligence Planning Systems*, vol. 2, 2002, pp. 281–290.

[97] G. Della Penna, D. Magazzeni, and F. Mercorio, "A universal planning system for hybrid domains," *Applied intelligence*, vol. 36, no. 4, pp. 932–959, 2012.

[98] L. Rosa, M. Cognetti, A. Nicastro, P. Alvarez, and G. Oriolo, "Multi-task cooperative control in a heterogeneous ground-air robot team," *IFAC-PapersOnLine*, vol. 48, no. 5, pp. 53–58, 2015.

[99] D. S. S. Miranda, L. E. de Souza, and G. S. Bastos, "A rosplan-based multi-robot navigation system," in *Proceedings of the Latin American Robotic Symposium*. IEEE, 2018, pp. 248–253.

[100] S. Zhang, Y. Jiang, G. Sharon, and P. Stone, "Multirobot symbolic planning under temporal uncertainty," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 501–510.

[101] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational IntelligenCe Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[102] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5561–5570.

[103] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 684–699.

[104] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3668–3678.

[105] L. Ma, Z. Lu, and H. Li, "Learning to answer questions from image using convolutional neural network," in *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, 2016.

[106] M. Garnelo, K. Arulkumaran, and M. Shanahan, "Towards deep symbolic reinforcement learning," *arXiv preprint arXiv:1609.05518*, 2016.

[107] M. Asai and A. Fukunaga, "Classical planning in deep latent space: Bridging the subsymbolic-symbolic boundary," in *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, 2018.

[108] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," *arXiv preprint arXiv:1904.12584*, 2019.

[109] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in *Proceedings of the IEEE 7th International Conference on Industrial and Information Systems*. IEEE, 2012, pp. 1–5.

[110] K. M. Wurm, C. Dornhege, B. Nebel, W. Burgard, and C. Stachniss, "Coordinating heterogeneous teams of robots using temporal symbolic planning," *Autonomous Robots*, vol. 34, no. 4, pp. 277–294, 2013.

[111] I. Jang, H.-S. Shin, A. Tsourdos, J. Jeong, S. Kim, and J. Suk, "An integrated decision-making framework of a heterogeneous aerial robotic swarm for cooperative tasks with minimum requirements," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 6, pp. 2101–2118, 2019.

[112] J. C. Reis, P. U. Lima, and J. Garcia, "Efficient distributed communications for multi-robot systems," in *Proceedings of the Robot Soccer World Cup*. Springer, 2013, pp. 280–291.

[113] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 2137–2145.

[114] K. Himri, P. Ridao, N. Gracias, A. Palomer, N. Palomeras, and R. Pi, "Semantic slam for an auv using object recognition from point clouds," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 360–365, 2018.

[115] L. Li, Z. Liu, Ü. Özgïner, J. Lian, Y. Zhou, and Y. Zhao, "Dense 3d semantic slam of traffic environment based on stereo vision," in *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2018, pp. 965–970.

[116] M. Mao, H. Zhang, S. Li, and B. Zhang, "Semantic-rtab-map (srm): A semantic slam system with cnns on depth images," *Mathematical Foundations of Computing*, vol. 2, no. 1, pp. 29–41, 2019.

[117] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 375–383.

[118] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

[119] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 639–646.

[120] C. Dornhege, A. Hertle, and B. Nebel, "Lazy evaluation and subsumption caching for search-based integrated task and motion planning," in *Proceedings of the International Conference on Intelligent Robots and Systems workshop on AI-Based Robotics*, 2013.

[121] L. Manso, P. Bustos, R. Alami, G. Milliez, and P. Núnez, "Planning human-robot interaction tasks using graph models," in *Proceedings of International Workshop on Recognition and Action for Scene Understanding*, 2015, pp. 15–27.

[122] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[123] F.-C. Ghesu, B. Georgescu, Y. Zheng, S. Grbic, A. Maier, J. Hornegger, and D. Comaniciu, "Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 176–189, 2019.

[124] Q. Zhang and D. Sornette, "Learning like humans with deep symbolic networks," *arXiv preprint arXiv:1707.03377*, 2017.

[125] Q. Liao and T. Poggio, "Object-oriented deep learning," Center for Brains, Minds and Machines (CBMM), Tech. Rep., 2017.

[126] J. Moon and B. Lee, "Scene understanding using natural language description based on 3d semantic graph map," *Intelligent Service Robotics*, vol. 11, no. 4, pp. 347–354, 2018.

초 록

인간과 이종 로봇 간의 협업은 높은 유연성과 적응력을 보일 수 있다는 점에서 제조업에서 필드 로보틱스까지 다양한 분야에서 필연적이다. 특히, 서로 다른 능력을 지닌 로봇들과 인간으로 구성된 하나의 팀은 넓고 정형화되지 않은 공간에서 서로의 능력을 보완하며 복잡한 임무 수행을 가능하게 한다는 점에서 큰 장점을 갖는다. 효율적인 한 팀이 되기 위해서는, 팀의 공통된 목표 및 각 팀원의 현재 상황에 관한 정보를 실시간으로 공유할 수 있어야 하며 함께 의사 결정을 할 수 있어야 한다. 이러한 관점에서, 자연어를 통한 의미론적 환경 이해는 인간과 서로 다른 로봇들이 모두 이해할 수 있는 형태로 환경을 인지한다는 점에서 가장 필수적인 요소이다. 또한, 우리는 자연어 기반 환경 이해를 통해 네트워크 혼잡을 피함으로써 획득한 정보의 신뢰성을 높일 수 있다. 특히, 대량의 센서 데이터 전송에 의해 네트워크 대역폭이 증가하고 통신 QoS (Quality of Service) 신뢰도가 감소하는 문제가 빈번히 발생하는 필드 로보틱스 영역에서는 의미론적 환경 정보인 자연어를 전송함으로써 통신 대역폭을 감소시키고 통신 QoS 신뢰도를 증가시킬 수 있다. 본 학위 논문에서는 환경의 의미론적 이해 기반 인간 로봇 협동 방법에 대해 소개한다. 먼저, 로봇의 지도 작성 알고리즘을 통해 획득한 그래프 지도를 이용하여 자연어 문장과 검출한 객체 및 각 객체 간의 관계를 자연어 단어로 표현하는 그래프를 생성한다. 그리고 자연어 처리 결과를 이용하여 인간과 다양한 로봇들이 함께 협업하여 임무를 수행할 수 있도록 하는 프레임워크를 제안한다.

본 학위 논문은 크게 그래프를 이용한 의미론적 환경 이해와 의미론적 환경 이해를 통한 인간과 이종 로봇 간의 협업 방법으로 구성된다. 먼저, 그래프를 이용한

의미론적 환경 이해 부분에서는 의미론적 그래프 지도를 이용한 새로운 자연어 처리 방법에 대해 소개한다. 의미론적 그래프 지도 작성 방법은 로봇의 환경 인지 측면에서 많이 연구되었지만 이를 이용한 자연어 처리 방법은 거의 연구되지 않았다. 반면 컴퓨터 비전 분야에서는 이미지를 이용한 환경 이해 연구가 많이 이루어졌지만, 연속적인 장면들은 다루는데는 한계점이 있다. 따라서 우리는 그래프 스펙트럼 이론에 기반한 그래프 컨볼루션과 그래프 축소 레이어로 구성된 그래프 컨볼루션 신경망 및 순환 신경망을 이용하여 그래프를 설명하는 문장을 생성한다. 제안한 방법은 기존의 방법들보다 한 장면에 대해 향상된 성능을 보였으며 연속된 장면들에 대해서도 성공적으로 자연어 문장을 생성한다.

최근 딥러닝은 자연어 기반 환경 인지에 있어 급속도로 큰 발전을 이루었다. 하지만 인과 추론, 유추적 추론, 임무 계획과 같은 높은 수준의 프로세스에는 적용이 힘들다. 반면 임무를 수행하는 데 있어 각 에이전트의 능력에 맞게 행위들의 순서를 계산해주는 상징적 접근법(symbolic approach)은 추론과 임무 계획에 있어 뛰어난 성능을 보이지만 인간과 로봇들 사이의 의미론적 정보 공유 방법에 대해서는 거의 다루지 않는다. 따라서, 인간과 이종 로봇 간의 협업 방법 부분에서는 딥러닝 기법들과 상징적 플래너(symbolic planner)를 연결하는 프레임워크를 제안하여 의미론적 이해를 통한 인간 및 이종 로봇 간의 협업을 가능하게 한다. 우리는 의미론적 주변 환경 이해를 위해 이전 부분에서 제안한 그래프 기반 자연어 문장 생성을 수행한다. PDDL 플래너와 JENA-TDB는 각각 임무 계획 및 정보 획득 저장소로 사용한다. 제안한 방법의 효용성은 시뮬레이션을 통해 두 가지 상황에 대해서 검증한다. 하나는 동적 환경에서 임무 실패 상황이며 다른 하나는 넓은 공간에서 객체를 찾는 상황이다.

주요어: 인지 로보틱스, 의미론적 환경 이해, 3D 환경 그래프, 임무 계획, 인간-로봇 협업, 자연어 처리

학번: 2014-22559

감사의 글

지난 대학원 생활을 무사히 마무리할 수 있었던 것은 저를 지지하고 응원해 주셨던 많은 분의 도움 덕분입니다. 이 글을 통해 모든 분께 감사 인사드립니다. 그 도움 하나하나가 헛되지 않게 앞으로 사회에 나가서도 항상 감사하며 많은 사람에게 도움이 될 수 있는 사람이 되도록 하겠습니다.

교수님들께

먼저 부족한 저에게 기회를 주시고 지속적인 격려와 아낌없는 조언을 해주신 이범희 교수님께 진심으로 깊이 감사드립니다. 제가 이만큼 학문적으로나 인격적으로 성장할 수 있었던 것은 교수님의 가르침과 사랑 덕분입니다. 졸업 후에도 교수님께서 주신 가르침과 연구실에서의 소중하고 값진 경험을 바탕으로 더 단단하고 지속해서 발전할 수 있는 사람이 될 수 있도록 하겠습니다. 바쁘신 일정 중에도 귀한 시간 내어 주신 교수님들께 감사 인사드립니다. 먼저 저의 박사학위 논문 심사위원장을 맡아주신 조동일 교수님께 감사드립니다. 교수님의 조언처럼 앞으로도 더 성장할 수 있도록 노력하겠습니다. 심사를 위해 바쁘신 일정 중에서도 시간을 내어 주신 심형보 교수님께 감사드립니다. 교수님 덕분에 연구에서 부족한 부분을 보완할 수 있었습니다. 넓은 식견으로 심사 과정 동안 더 좋은 논문이 만들어질 수 있도록 여러 관점에서 함께 고민해 주신 윤성로 교수님께 감사드립니다. 빅데이터 및 딥러닝 분야에서 활발하게 연구 활동을 펼치는 교수님을 심사위원으로 모시게 되어 영광이었습니다. 마지막으로 심사를 위해 먼 거리를 와주신 박재병 교수님께 감사드립니다. 교수님 덕분에 무사히 심사를 마칠 수 있었습니다.

연구실 선·후배님들께

연구실 생활을 함께한 선배님 및 후배님들께 고마움을 전하고 싶습니다. 제가 처음으로 대학원에 입학하였을 때 잠깐이지만 함께 연구실 생활을 한 두진이 오빠, 가끔이지만 오빠를 뵐 때마다 참 반갑습니다. 항상 연구실 방장일 것만 같았던 규호 오빠, 오빠의 세심한 배려와 조언들은 낯설었던 연구실 생활에 적응하는 데 큰 도움이 되었습니다. 크고 작은 일에 대해 함께 이야기 나누었던 시간이 그립습니다. 항상 웃음으로 대해 주신 승환이 오빠, 처음에는 선뜻 다가가기 힘들었지만, 지금은 제일 편한 선배님입니다. 어떤 일이든 최선을 다하는 오빠의 모습을 보면서 항상 많이 배웁니다. 어떻게 연구를 시작하여야 하는지, 발표 자료는 어떻게 만들어야 좋은지, 논문은 어떻게 작성하여야 하는지 정말 많은 도움을 준 재도 오빠, 항상 고마운 마음으로 가득한데 잘 전달하지 못한 것 같아 마음이 무겁습니다. 항상 고마운 마음 잊지 않고 있어요. 저에게 먼저 다가와서 말 걸어주었던 다정한 훈수 오빠, 항상 후배를 챙겨주고자 하는 따스함이 의지가 되었습니다. 언제나 유머를 잃지 않는 정현이 오빠, 제가 입학해서부터 지금까지 다방면으로 잘 챙겨주셔서 고맙습니다. 오빠 덕분에 제 연구를 시작하고 연구실 생활을 잘 마무리 할 수 있었습니다. 앞으로도 잘 부탁드려요. 어떤 일이든 적극적으로 도움을 주었던 원석이 오빠, 제가 어떻게 연구를 해야 하는지 고민이 깊었을 때 오빠의 격려 덕분에 힘을 낼 수 있었습니다. 세심하고 배려가 깊은 현기 오빠, 졸업 전에 맡은 일이 많아서 연구와 다양한 일을 완벽하게 병행하면서도 저희를 챙기는 오빠의 모습을 보면서 대단하다고 생각했습니다. 오래는 같이 지내지 못했지만 알게 모르게 든든한 지웅이, 좀 더 많은 이야기를 나눌 기회가 있었으면 하는 아쉬움이 남습니다. 멀리에서 유학 생활로 고생하고 있을 지훈이, 한국으로 돌아오기 전에 한 번 더 찾아갈 기회가 오면 좋을 것 같습니다. 먼저 다가와 말 걸어주고 어떻게 연구를 진행하고 있는지 관심을 주었던 현우, 적절한 조언으로 연구의 방향을 잡아주어 논문의 완성도가 크게 높아질 수 있었습니다. 이 글을 통해 고마운 마음 전하고 싶습니다. 어떤 말이든 편하게 이야기할 수 있었던 원영이, 덕분에 조금 더 즐거운 연구실 생활을 할 수 있었습니다. 매사에 누나라고 세심하게 배려해 준 마음 잊지 않겠습니다. 연구실 생활 하는 동안 가장 의지 되었던

한준이 오빠, 덕분에 편하게 연구실 생활을 마칠 수 있었습니다. 오빠가 없었으면 졸업하기 힘들었을 것 같습니다. 먼저 연구실을 떠난 진원이, 졸업 후에 회사에서도 인정받는 모습이 자랑스럽습니다. 동시에 함께 연구실에서 생활했던 시간이 그립기도 합니다. 마지막으로 제 옆자리에 있었던 현일이, 다양하고 많은 책을 읽던 준혁이, 동갑이지만 동생이었던 호웅이에게도 감사 인사 전하고 싶습니다. 이렇게 좋은 사람들과 함께 연구실 생활을 할 수 있어서 행복했습니다. 모두 각자의 위치에서 바쁘겠지만 앞으로도 지속해서 연락하고 좋은 인연 이어갈 수 있었으면 좋겠습니다.

부모님 및 가족에게

항상 전폭적인 지지 및 지원을 해주신 부모님 및 가족들에게 감사의 인사 전합니다. 먼저 아버지, 어머니의 딸로 태어나 사랑받을 수 있었던 것이 가장 큰 행운이라고 생각합니다. 항상 성실하게 매사에 최선을 다하는 아버지와 어머니 모습을 보며 자라왔기에 지금의 제가 있지 않나 싶습니다. 한결같이 믿어주고 지지해 주셔서 감사드립니다. 아버지, 어머니는 지금까지 저의 자랑이었고 앞으로는 제가 아버지, 어머니의 자랑이 될 수 있도록 하겠습니다. 사랑합니다. 이제는 10년을 가까이 함께한 이모부와 이모, 저에게는 부모님과 다름없습니다. 이모부와 이모가 계셨기에 지금까지 학업을 잘 마무리할 수 있었습니다. 항상 좋은 말씀 많이 해주시고 힘을 주셔서 감사드립니다. 아닌 척하지만 듬직한 우리 장혁이, 요즘 새로운 전공 공부에 매우 힘들겠지만, 누구보다도 잘 해낼 거라고 믿습니다. 동생이 있어 정말 든든합니다. 공부도 잘하는데 누가 보아도 정말 예쁜 우리 나영이, 속 깊은 모습을 볼 때마다 언제 이렇게 자랐나 놀랍습니다. 앞으로의 우리 사랑하는 나영이의 재미있고 새로운 경험으로 가득할 대학 생활을 응원합니다. 저보다 12살이나 어린 동생이지만 부족한 언니를 챙겨주고 귀여워해 주는 최고 예쁜 우리 다현이, 지금도 충분히 잘하고 있지만, 하루하루 더 열심히 노력하는 모습을 보면 정말 장하고 사랑스럽습니다. 옆에서 조금이라도 도움이 될 수 있도록 제가 최선을 다해 도울 수 있도록 하겠습니다. 앞으로 우리 장혁이, 나영이, 다현이 모두 함께 서로 의지하면서 고민은 나누며 행복하게 잘 지냈으면 좋겠습니다. 제가 더 든든한 누나, 언니가 될 수 있도록 노력하겠습니다. 마지막으로 다시 저를 기억해 주셨으면 하는 할머니, 항상 손주들의 좋은 소식을

기다리시는 외할머니, 외할아버지, 항상 감사하고 건강하세요. 가족들이 있었기에 학업을 잘 마칠 수 있었다는 것을 잘 알고 있습니다. 가족 모두에게 감사드리고, 다시 한번 고맙다는 말 전하고 싶습니다.