ISSN : 1613-0073
CEUR-WS: Vol-2487
urn:nbn:de:0074-2487-8

co-located with IROS MACAU 2019

2019 IEEE/RSJ International Conference on Intelligent Robots and Systems
November 4-8, 2019
The Venetian Macao, Macau, China
www.iros2019.org

JOINT PROCEEDINGS

1st International SDMM 2019 Workshop

Semantic Descriptor, Semantic Modeling and Mapping for Humanlike Perception and Navigation of Mobile Robots toward Large Scale Long-Term Autonomy

&

3rd International AnSWeR 2019 Workshop

Applications of Knowledge Representation and Semantic Technologies in Robotics

Organizer:

co-Organizers:

IROS 2019 Organizers:

Joint Proceedings of

SDMM 2019
The 1st International Workshop on the Semantic Descriptor, Semantic Modeling and Mapping for Humanlike Perceptionand Navigation of Mobile Robots toward Large Scale Long-Term Autonomy

---

and

AnSWeR 2019
The 3rd International Workshop on the Applications of Knowledge Representation and Semantic Technologies in Robotics

---

co-located with

The International Conference on Intelligent Robots and Systems (IROS 2019)
Macau, China, November 4-8, 2019

Volume Editors

SDMM19

Tae-Yong Kuc,
Sumaira Manzoor

Sungkyunkwan University
South Korea
(tykuc, sumaira11)@skku.edu

AnSWeR19

Ilaria Tiddi

Vrije Universiteit Amsterdam (NL)
i.tiddi@vu.nl

Masoumeh Mansouri

Birmingham University (UK)
m.mansouri@bham.ac.uk

Emanuele Bastianelli

Heriott-Watt University (UK)
e.bastianelli@hw.ac.uk

Amelie Gyrard

Wright State University, USA)
amelie@knoesis.org

Preface

This joint volume of proceedings gathers papers from the 1$^{\mathbf{st}}$ International Workshop on the Semantic Descriptor, Semantic Modelingand Mapping for Humanlike Perceptionand Navigation of Mobile Robots toward Large Scale Long-Term Autonomy (SDMM19) and the 3$^{\mathbf{rd}}$ International Workshop on the Applications of Knowledge Representation and Semantic Technologies in Robotics (AnSWeR19). SDMM19 held on November 8, 2019 and AnSWeR19 held on November 4, 2019 during the International Conference on Intelligent Robots and Systems (IROS 2019) in Macau, China.

Preface

The 1st International Workshop on the Semantic Descriptor, Semantic Modelingand Mapping for Humanlike Perceptionand Navigation of Mobile Robots toward Large Scale Long-Term Autonomy (SDMM19)

A big portion of our common surroundings was created by humans, for humans. Over the centuries, we shaped the environments surrounding us according to our own conceptions and convenience. With the growing need for robots that can perform tasks on those large-scale dynamic environments, it is paramount that those robots can understand the World in the same fashion as humans do. Being able to reason and perform high-level tasks, with human-like learning and cognitive skills that can enhance their task planning and fast adaptation to highly dynamic surroundings, while also storing and utilizing past experiences are crucial skills for the next generation of robots. However, the current tools still mostly focus on machine-centric environment modeling, which reiterates the need of a new human-like environment and knowledge model.

This workshop will introduce semantic descriptor, semantic modeling and mapping framework for humanlike high-level perception and navigation of mobile robots toward large scale long-term autonomy in global dynamic environment. Based on the understanding of visual sensory information processing of human from cognitive science and efficient and flexible brain GPS model from neuroscience research and physiology*, triplet ontological semantic model (TOSM) has been addressed and used not only in object detection and place recognition but in generating layered semantic object-feature-topology-metric maps. With the framework idea and its extension to AI algorithms, a set of attractive topics will be presented and discussed in the workshop including semantic analysis and semantic information processing with semantic descriptors, space-time independent object detection and place recognition, AI based long-term planning and robot localization, and TOSM based robust semantic SLAM for global long-term autonomy.

The workshop is also aiming at providing a chance to robotic researchers, engineers, and students to review, evaluate, and advance a formal semantic modeling and mapping framework for humanlike high-level environment perception and navigation of robot. The topics covered by this workshop are relevant to the audience not only from robotic researchers but computer vision scientists who study place recognition and localization under visual appearance changes due to weather condition and time.

Topics of Interest

- One entry in the list AI Planning for long-term mission (AI Planning)
- Triplet ontological semantic model(TOSM) for workspace modeling and mapping (Semantic Modeling)
- Semantic analysis and semantic descriptors for object detection and place recognition (Semantic Descriptor, Object Detection, Place Recognition)
- Learning semantic descriptors and object detection by using deep neural network (Semantic Descriptor, Deep Neural Network)
- Global-local semantic SLAM for large scale long-term autonomy (Semantic SLAM)

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Preface

The 3rd International Workshop on the Applications of Knowledge Representation and Semantic Technologies in Robotics (AnSWeR19)

This volume gathers papers from the 3rd International Workshop on Applications of Knowledge Representation and Semantic Technologies in Robotics (AnSWeR19), which was held on November 4th, 2019 during the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2019) in Macau, China.

As robots are slowly approaching our everyday lives, they will need to expose an increasing capability to deal with different sources of knowledge about the world, in order to accomplish complex tasks based on Planning, Computer Vision, Natural Language Processing and many other techniques.

While the problem of enabling robots to use available sources of knowledge has attracted attention relatively recently in the robotics community, the Knowledge Representation community has been studying techniques to model, integrate and exploit heterogeneous sources of knowledge for a long time.

The aim of the workshop is to promote and strengthen the dialogue between the Knowledge Representation and Robotics communities that are working on connected, overlapping topics, and to find answers to common research questions. The stimulated debate served as a background in fostering the application of Knowledge Representation techniques in Robotics, and in highlighting Robotics as a fertile application field for the KR community.

Three papers were accepted in this third edition of AnSWeR; all of these are presented in this volume. Additionally, the workshop hosted 5 invited talks around the combination of KR and Robotics, namely :

1. Lars Kunze (UK) : Autonomous Robots in a Connected World;
2. Todor Stoyanov (SW) : Semantic mapping for robots and by robots: the role of high-level information
3. Yuke Zhu (US) : Learning How-To Knowledge from the Web
4. Mathieu d'Aquin (IE) : Virtualized knowledge for robot understanding
5. Vera Ragavan (US) : An overview of IEEE 1872.2 WG "Autonomous Robotics Ontology Progress" and "Towards an Ontology driven Design and Development Process"

The editors would like to thank all the authors for their insightful contributions to AnSWeR. A special thank goes also to members of the program committee, which ensured a high quality standard for the workshop through their review assessment.

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Organizing Committee

SDMM19

Main Organizers

Prof. Tae-Yong Kuc

School of Electronics and Electrical Engineering
College of Information and Communication Engineering
SungKyunKwan University
Serburo 2066 Suwon, 16419 South Korea
Phone: +82-31-290-7137
Email: tykuc@skku.edu

Co Organizers

Dr. Stephanie Lowry

Center for Applied Autonomous Sensor Systems
School of Science and Technology
Örebro University
70219 Örebro Sweden
Email: stephanie.lowry@oru.se

Prof. Fei Qiao

Deparment of Electronics Enginnering
Tsinghua University
100084 Beijing, P.R. of China
Email: qiaofei@tsinghua.edu.cn

Dr. Sang-Hoon Ji

Convergence Research Center, Robot R&D Group
Korea Institute of Industrial Technology
Ansan 426-791 South Korea
Phone: +82-31-8040-6363
Email: robot91@kitech.re.kr

Organizing Committee

AnSWeR19

Dr. Ilaria Tiddi

Vrije Universiteit Amsterdam
Netherlands
Email:i.tiddi@vu.nl

Dr. Masoumeh Mansouri

Birmingham University
UK
Email: m.mansouri@bham.ac.uk

Dr. Emanuele Bastianelli

Heriott-Watt University
UK
Email: e.bastianelli@hw.ac.uk

Dr. Amelie Gyrard

Wright State University
USA
Email: amelie@knoesis.org

Contents

SDMM19

A Novel Semantic SLAM Framework for Humanlike High-Level Interaction and Planning in Global Environment 10

The Semantics of Context: The Role of Interpretation and Belief in Visual Localization for Robots 22

Dense-Loop: A Loop Closure Detection Method for Visual SLAM using DenseNet Features 27

Learning Safety-Aware Policy with Imitation Learning for Context-Adaptive Navigation 38

Guidance of Mobile Robot Navigation in Urban Environment Using Human-Centered Cloud Map 48

Semantic Information-based Reliable Autonomous Navigation in Wide Space 53

Towards Explanations of Plan Execution for Human-Robot Teaming 58

Mental Simulation for Autonomous Learning and Planning Based on Triplet Ontological Semantic Model 65

Combining Semantic Modeling and Deep Reinforcemt Learning for Autonomous Agents in Minecraft 74

Contents

AnSWeR19

How Does a Robot Speak? About the Man-Machine Verbal Interaction 78

Hybrid Question Answering System based on Natural Language Processing and SPARQL Query 94

Auto-Perceptive Reinforcement Learning (APRiL) 103

A Novel Semantic SLAM Framework for Humanlike High-Level Interaction and Planning in Global Environment

Sumaira Manzoor, Sung-Hyeon Joo, Yuri Goncalves Rocha, Hyun-Uk Lee, Tae-Yong Kuc
College of Information and Communication Engineering,
Sungkyunkwan University, South Korea
{sumaira11,sh.joo, yurirocha, zlshvl36, tykuc}@skku.edu

Abstract

In this paper, we propose a novel semantic SLAM framework based on human cognitive skills and capabilities that endow the robot with high level interaction and planning in real-world dynamic environment. Two-fold strengths of our framework aims at contributing: 1) A semantic map resulting from the integration of SLAM with the Triplet Ontological Semantic Model (TOSM); 2) Human-like robotic perception system that is optimal and biologically plausible for place and object recognition in dynamic environment proposing semantic descriptor and CNN .We demonstrate the effectiveness of our proposed framework using mobile robot with Zed camera (3D sensor) and a laser range finder (2D sensor) in real-world indoor environment. Experimental results demonstrate the practical merit of our proposed framework.

1 Introduction

Building the autonomous mobile robot with human-like intelligence for semantic map construction and cognitive vision-based perception are two the most significant challenges for long-term planning and high-level interaction in indoor environment.

The problem to determine the appropriate method for building and maintaining the map that encodes both casual and world knowledges has become an active research area in the robotics. Many studies in the last decades have focused on spatial representation of the environment for building metric, topological and appearance-based maps. However, semantic mapping of environment for the robots has not been as intensively studied. The information provided by the conventional mapping approaches assists only in robot navigation while qualitative information about the structure of environment for task planning is not generated. For instance, metric map that contains geometric representation of the environment provides shape of the room without any semantic understanding to indicate whether it is office or lecture room. Our proposed framework tackles this issue by constructing the map that combines spatial representation with semantic knowledge of environment and provide autonomous navigation to robot for perform high-level task without human intervention in global dynamic environment.

The semantic interpretation of the environment also plays an essential role to improve the perception ability of the robot for performing real-world operations such as object and place recognition in more reliable and intelligent

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

manner. Nowadays, approaches for robotic perception range from traditional computer vision using handcrafted features to advanced deep learning with convolutional neural network or combination of both. However, these artificial vision algorithms have practical limitations to process in real time [bohg17]. Therefore, biologically plausible algorithms combined with analogies of artificial perception are getting the attention. Our proposed framework handles the current challenges by developing the effective solution that enables the robot with the potential of human-like vision for recognizing the objects and places using semantic perception.

The primary goal of our novel semantic framework is twofold for developing semantic perception system and endowing the robot to incrementally build a consistent semantic map while simultaneously determining its location within map.

Our proposed semantic SLAM framework makes an original contribution to three important research areas in robotics with the following characteristics:

- Human-like brain GPS system for building semantic maps with emphasis on qualitative description of robot's surrounding
- Human cognition based 1TOSM with deeper domain knowledge acquired by semantic, topological and geometric properties of the objects for providing the robot higher degree of autonomy and intelligence.
- Bio-inspired semantic perception system combined with object and place recognition that allows the robot to relate what it perceives using semantic descriptor

This paper is organized as follows. In Section II, we provide an extensive literature review of semantic mapping, semantic SLAM and perception system for autonomous mobile robot. In Section III, we explain the key features of our proposed framework with complete details of major components of TOSM and recognition model. In Section IV, we examine the significant effects of our proposed framework in simulated environment as an illustration of its contents. Finally, we conclude our work with future direction in Section V.

2 Related Work

We focus our review on studies of three major concepts, which we consider are the most closely related to our work: a) semantic SLAM b) ontology c) semantic perception for object and place recognition d) semantic descriptor.

2.1 Semantic SLAM

This section, gives the understanding of SLAM, explain semantic SLAM structure, its concepts and related work in this area.

A. Semantic Mapping

In the last few years, embedding the map with semantic information has become an active research area with the motivation of human-like robot interaction and understanding of the environment. High-level features in semantic map are used to model the human concepts about the objects, places and relationship between them [Capobianco15]. Semantic mapping has recently become the center of attraction in research community which divides the semantic mapping approaches into three groups based on object, appearance and activity [Pendleton17]. Object based semantic mapping [Vasudevan08]methods depends on the occurrence of key objects to perform object recognition and classification tasks by semantic understanding of environment. Appearance based semantic mapping approaches take sensor readings and interpret them for constructing semantic information of the environment. Some studies use geometric features [Burgard07] and vision fused with LIDAR data for world understanding and classification [Nüchter08] task. The activity based semantic mapping [Xie13] techniques use information of external activities (e.g. sidewalk verses roads) around the robot for semantic understanding and contextual classification of environment. These techniques are at their formative stage compared to other two semantic mapping methods.

B. Semantic SLAM: Concepts

The large number of concepts and relationship among them in real-world environment lead to several task-driven decisions which depends on the level of semantic organization and context of environment in which robot

performs its task. Literature review shows two major concepts of constructing semantic relationships [cadena16] based on the details and organization. The detail of semantic concept significantly affects the complexity of the problem at different levels. For example, a robot needs only coarse categories such as rooms, doors and corridors to perform a task "going form 1st room to 2nd room" while for the other task "pick up the glass" it needs to know finer categories such as table, glass or any other object. The semantic concepts are not limited because a single entity or object in real-world environment has many properties or concepts. For example, "moveable" and "sittable" are the properties of a chair while "movable" and "unsittable" are the properties of a table. Both table and chair have same class "Furniture". However, they share "moveable" property with different usability. So, this multiplicity of concepts is handled by Flat or hierarchical organization of properties

Semantic SLAM: Object/ Place Recognition

Semantic to the SLAM is included by using human-spatial concepts into the maps. Humans locate themselves by object centric concepts instead of metric information and they use reference points rather than global coordinates. The initial research into semantic mapping uses direct approach [Lowry16] with metric map segmentation built by traditional SLAM system into semantic concepts. An early work in [Sabourin10] develops a system for scene understanding via semantic analysis using image segmentation techniques and the SLAM algorithm is driven by object recognition using human spatial concepts. The work shows that semantic concepts are organized in in coarse to finer manner for indoor environment. An online semantic mapping framework [Pronobis12] of indoor environment combined with object observations such as shape, size, room's appearance that is built using three layers of reasoning to address the problem of detection and learning of novel properties and room categories for fully self-extendable semantic mapping. Data association problem also exists in metric and semantic SLAM when building a map of environment with large number of objects of the same or different class and scales. This problem is addressed in [Bowman17] by coupling geometric and semantic observations and taking the advantage of object recognition for providing meaningful scene interpretation with semantically labeled landmarks.

2.2 Ontology

In recent years, reducing the semantic gap using ontologies has been studied by many researchers. An early study [Durand07], has introduced an object recognition approach based on ontology and assigned the semantic meaning to objects by matching process between concepts and objects. The work in [Ji12], handles the robot task planning issues in domestic environment at the high symbolic level by combining classical AI approaches with semantic knowledge representation. Its framework is based on semantic knowledge ontology to represent robot primitive actions and description of environment. A study in [Riazuelo15], described the RoboEarth project using knowledge-based system to provide web and cloud services to multiple robots. Its semantic mapping system is based on visual SLAM mapping and ontology to describe the concepts, relations in maps and objects. A robotic system with advanced abilities leads to the complexity in its software development. A case study presented in [Saigol15] addresses this issue using an ontology as the central data store to process all information and showed that knowledge-base makes the robotic system easier to develop, modify and understand. In the last few years, a variety of approaches have been investigated to process the sensory information in dynamic world. Among them, OnPercept [Azevedo18] is a recent approach that is based on cognitive ontology for performing the HRI tasks by modeling the sensory information. A study [Lee18], proposes context query-processing framework using spatio-temporal context ontology for enabling the indoor service robots to adapt the dynamic change from the sensors in highly complex environment.

2.3 Perception

Perception system endows the robot to perceive and reason about its environment. The autonomous mobile robot can perform its complex tasks such as object and place recognition, collision avoidance, task planning, decision making, mapping, dynamic interaction, localization, and intelligent reasoning with high accuracy if perception information is carefully processed. A recent study has [Sünderhauf18] highlighted the fact that robotic perception is different from conventional computer vision perception because in computer vision image output is taken as information while a robotics perception system translates the image output from information into actions for taking decisions and actions in real world environment. Therefore, perception plays vital role for the success of goal-driven robotic system. However, despite this difference, robot perception incorporates the techniques from computer vision, and it is particularly evolving with the recent development in deep learning networks.

In real-world applications, endowing a robot with human like-perception for navigation is a challenging task that enables the robot to recognize scene and object when navigating through a dynamic complex environment and building a 3D map by observing the surrounding. Therefore, regardless of selected navigation system, object identification and place recognition play a vital role for environment representation and modeling.

A. Object Recognition

Reliable object recognition is an important and early step for a mobile robot to achieve its goal. Real time object recognition systems work in two stages: Offline and online. Offline stage aims at reducing the execution time without affecting system efficiency. Image pre-processing, feature extraction, segmentation and training processes are performed in this stage. Online stage runs the process in real time to ensure the high-level interaction between robot and its surrounding environment. Image retrieval, classification, object detection and recognition are the examples of few processes that are carried out at this stage.

A key issue in this context is the interaction with object of different shapes and sizes. Despite significant achievements and advent of digital camera, accurate object detection and recognition is still a challenging task when real-world environment is considered. The reasons for this difficulty are occlusions, complex object shapes, variations in geometric and photometric pose, noise and illumination changes.

Early efforts [Zou19] to handle this issue are based on template matching. Later approaches include statistical classifiers including SVM, Adaboost and neural networks. On the other hand, computationally simple and efficient approached based on local features such as scale-invariant descripts (e.g. SURF, SIFT), haarlike features also exist. However, the limitations of these methods include accuracy that depends on number of features that describe an image, segmentation that becomes highly complex in real world scenarios and not robust to relatively large affine transformations. In literature, its alternative is to use Object Action Complexes (OACs) [Petrick08] that combines the action, object and learning process to deal with the representational difficulties in diverse areas.

The perception-action relationship based on cognitive understanding has been explored in [Yan14] by linking both tasks through a memory component. In these studies, perception system uses three sensor modalities: vision, audio and touch and their data are passed to the memory module for generating the motor control signals and action unit translate them into robot responses. This intermediate process acts as robot's brain for improving the recognition task when mobile robot navigates in unknown environment. The study of attention based cognitive architecture in [Palomino16] uses the reasoning as a bound between perception and action. The core of this work is selection of active task based on the context data and accomplishment of task depends on the presence of specific element in the scene. However, object-based visual attention system still requires considerable efforts to accurately detect and categorize different objects.A recent study [Ye17] presents a vision system for object detection and recognition from a visual input in real time by computing motion, color, motion and shape cues and combining them in a probabilistic manner for assistive robots.

However, despite the vast analysis of existing perceptual systems for autonomous mobile robots, semantic recognition system remains to be addressed for robust object recognition in real-world scenario.

B. Place Recognition

Visual place recognition becomes very challenging when real-world scenario is concerned. Therefore, visual place recognition algorithms must endow the autonomous mobile robot to robustly handle the variation in visual environment that occur due to dynamic, geographical and categorical changes [Martinez17]. The visual appearance of places varies due to illumination changes (day and night), moving the furniture or different objects from one place to another. The same place (room or corridor) might look different in different viewpoints, despite sharing some common visual features. Humans can recognize a room (office or kitchen) because of their ability to build categorical models of places. However, it is difficult for the robot to recognize the rooms based on their distinctive features and categories.

Literature review [Ullah08] shows that contextual understanding of the place is very important for autonomous mobile robot to effectively perform its task. A mobile robot can effectively interact with its environment if it recognized the place and have a functional understanding of area

2.4 Semantic Descriptor

There have been few empirical investigations in recognizing the objects that have semantic similarities in their shapes. A recent study [Tasse16], address this challenge and computes the semantic similarities between shapes,

images and depth maps using semantic based descriptors. The central idea is to combine labeled 3D shapes with semantic information in their labels for generating semantic-based 3D shape descriptor. An early study [Zen12], uses enhanced semantic descriptors for complex video scene understanding by embedding semantic information in the visual words. Recent developments in robot localization and mapping approaches have heightened the need to use semantic descriptors for robot localization and mapping. A seminal study in [Panphattarasap18], uses 4-bit binary semantic descriptor (BSD) for robot localization in 2-D map and performs semantic matching. The semantic features such as gap between buildings and road junctions are detected using CNN in urban environment. The purpose of BSD is to endow the robot with ability akin to human map reading.

3 Framework

Our proposed framework adds semantic techniques to SLAM to cope with the challenges in dynamic environment by providing the robot advanced perception that is closer to human vision and improving the world understanding capabilities of the robot for carrying out high-level navigation task in complex unstructured environments. Our framework provides a closer representation with global environment by defining the Triplet Ontological Semantic Model (TOSM) in which relations between the concepts are described for explaining semantic interoperability of environment.

3.1 TOSM: Triplet Ontological Semantic Model

We accelerate the implementation of cognitive system in autonomous mobile robot by developing Triplet Ontological Semantic Model (TOSM) which is based on cognitive process of human perception and brain GPS model from neuroscience research and physiology. The main characteristics of TOSM are:

- To endow the robot with semantic mapping of environment based on cognitive architecture modeling
- To define the relations between domain concepts (knowledge), their attributes (properties) with high-level of abstraction and rules to reason based on the task and the environment
- To model the sensory information for performing the task planning

Our TOSM approach consisting of three major components for effective representation of domain knowledge and information retrieval in indoor environment is shown in Fig 1. Unique characteristics of these three components represent relationship information with different objects that have spatial and non-spatial properties for performing a specific task in overall robotic environment. The spatial properties represent the concepts of position, shape and size of the objects in robotic environment while the non-spatial properties determine the object category. We describe complete domain knowledge using spatial representation of the objects. Our proposed TOSM approach endows the robot to semantically map the objects and their positions in unexplored environment by defining explicit, implicit and symbolic models, shown in Figure Figure 1

A. Explicit Model

Explicit model specifies the spatial representation of the entities such as shape of an object and its position in the domain (global environment) by extracting all the geometrical features of that object and retrieving its physical information from sensors.

B. Implicit Model

Implicit model describes the behavior of the robot and series of actions such as robot navigation to perform a task. This spatial representation also defines the intrinsic relations between the entities, gives the semantic interpretation of environment which cannot be obtained using sensors and processes the fuzzy information to provide the effective interaction of the mobile robot with its surrounding along with planning capabilities. Introducing this model in our framework also enables the robot to take high-level decision by understanding the semantic concepts that constitute task success, such as it allows the robot to interpret the semantics of automatic door by understanding its salient events that auto-door opens and closes automatically, on sensing the approach of a person.

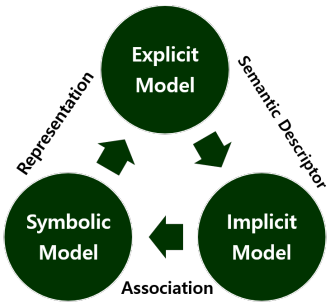


Figure 1: Triplet Ontological Semantic Model (TOSM)

C. Symbolic Model

We use symbolic model to encode the domain knowledge for describing semantic descriptions, sequence of actions and complex capabilities of our environment in language-oriented way. Robot uses this knowledge through relations that are represented by the links between existing entities. Based on the integrated components of implicit, explicit and symbolic models, the TOSM approach coexist in SLAM allows the robots to perceive, learn, understand and interact with the surroundings based on geometric and semantic information.

3.2 TOSM on demand database

We design robot-mounted on-demand database to construct semantic model of the environment for providing the robot a semantic mapping and perception closer to human cognitive skills using TOSM. Our TOSM on-demand database approach has three main practical advantages:

- It eliminates the demand to store several different maps
- It generates the maps only when they are required for a robot to perform the assigned task in global dynamic environment
- It enriches the database semantically by adding conceptual meaning to data and relationships

We store, environmental and behavioral information together with robot knowledge and map data in on-demand database. Robot uses cloud database to plan the behavioral actions and on-demand database to builds a dynamic driving map according Figure 1: Triplet Ontological Semantic Model (TOSM)to assigned task in operating environment. If the robot needs additional information to download from network or cloud database for performing a specific task, this information is also merged with the robot's current knowledge and on-demand database is concurrently updated. The on-demand database of environment based on TOSM describes the semantics of the domain with the set of relations. We have developed it using the protégé tool to explicitly represent the class hierarchy for each individual. Individuals, also called instances, are defined to represent a specific object in a class. For instance, automatic door is an individual of 'Door' class, as shown in Figure 2(a). We describe our ontological model by creating individuals (instances) in corresponding classes, connecting them with typed literals and defining relationships between objects of different classes. TOSM for on-demand database is composed of three main components: classes, object properties and data properties.

A. Classes

We use classes to describe the concepts using collections or types of objects that share common properties in indoor environment. Our ontological model consists of five classes: Map, MathematicalStructure, Time, Behavior and EnvironmentElement. Ecah class represents an abstract group of objects that belongs to the specific class. TOSM allows the classes to have single inheritance (one parent) and multiple inheritance. For example, subclasses of object, Occupant, Robot and Place in EvnirnmentElement class have single inheritance while AutomaticDoor class has multiple inheritance. Thus, all the properties of parents' classes (Door, Object and EnvironmentElement) are inherited by child class (AutomaticDoor). TOSM uses subclasses to represent the concepts more specifically than super classes. Figure 2(a) also shows that we have developed our class hierarchy with the systematic top-down view of domain in which we define the most general concepts of an entity in high-level (superclass) and more specific concepts in low-level (subclass).

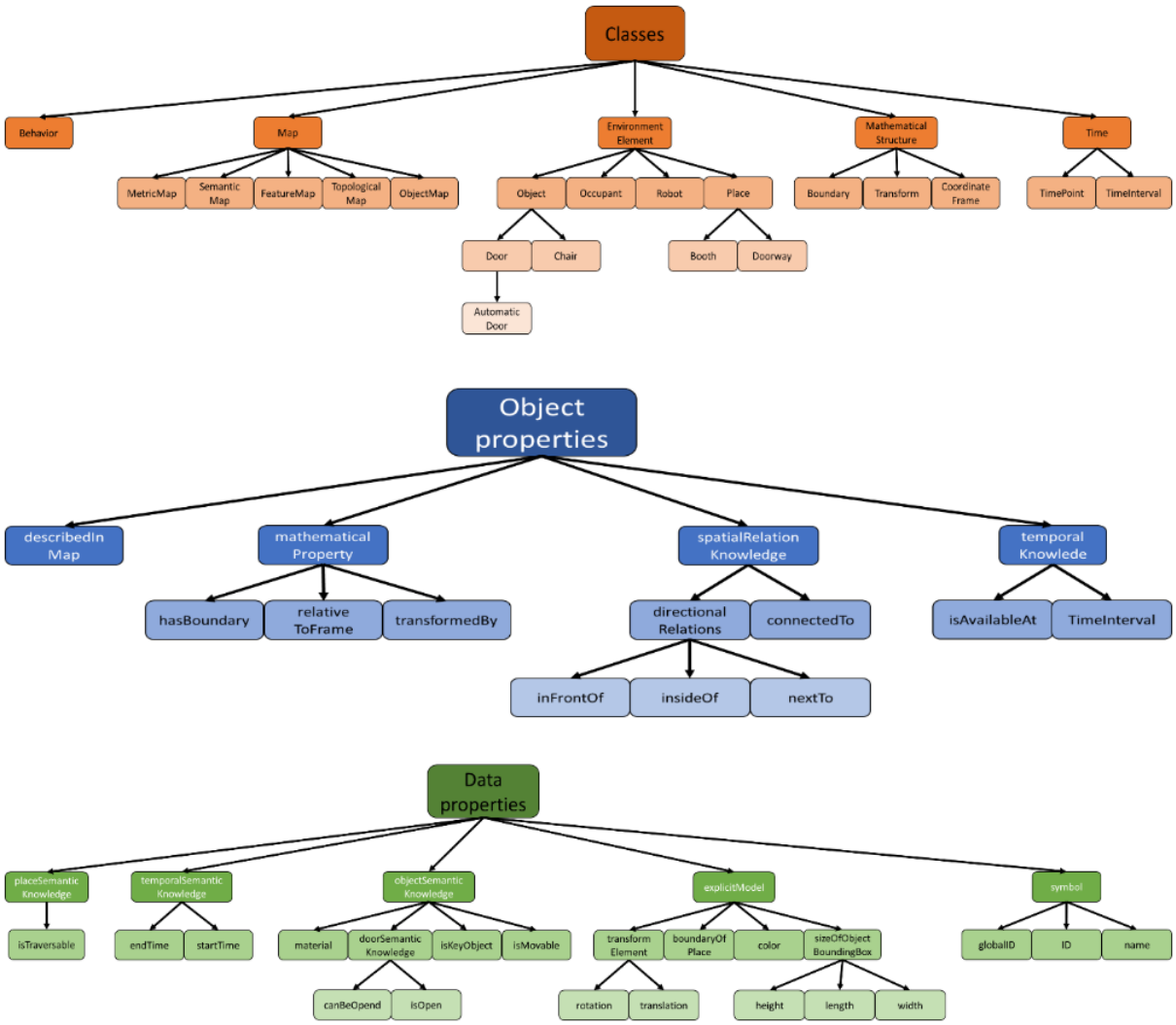


Figure 2: TOSM Properties for on demand database. (a) Class Properties; (b) Object Properties; (c) Data Properties

B. Object Properties

These properties explain the relationship between the classes based on their instances. The category of object and set of properties determine the type of relationship between them. Figure 3 shows the expression of 3D geometric relation between two classes: "Room1 hasBounday Boundary1" in which an object property "hasBoundary" links the individual "boundary1" of MathematicalStructure class to the individual "room1" of "EnvironmentElement" class. This geometric relation is inferred from visual perception and semantic map.

We divide the object properties into describedInMap, mathemeticalProperty, spatialRelationKnowledge and temporalK. Figure 2(b) shows that mathematicalProperty includes hasBoundary,relativeToFrame and transformedBy, whereas spatialRelationKnowledge includes connectedTo and directionalRelations which is divided into inFrontOf, insideOf and nextTo. Finally, temporalKnowledge include isAvailableAt and Timeinterval properties

C. Data Properties

These properties specify object parameters or typed literal, also called datatype (string, int, float). We retrieve the individuals by connecting them with the specified literal values using placeSemanticKnowledge, temporalSemanticKnowledge, objectSemanticKnowledge , expliciitModel and symbol that are defined as data properties in

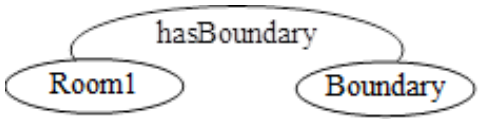


Figure 3: Geometric Relation between Two Classes

our ontological model, as shown in Figure 2(c).

3.3 Semantic descriptor-based Learning and Recognition

Our proposed framework introduces real-time and near-perfect object detection and place recognition approaches to mimic human visual system using semantic descriptor-based learning. The overview of our recognition model inspired by human visual cortex and semantic descriptor is illustrated in Figure 4.

When autonomous mobile robot explores complex indoor environment to perform a task, the perception module recognizes the objects and places by extracting data from sensors and retrieving from on-demand TOSM database. It continuously updates symbolic state of the task based on semantic information of newly obtained data from sensors and adds the implicit data about novel objects and places by identifying their classes in knowledge base.

Our framework allows open-ended learning that enables the robot to adapt to new environment by acquiring the knowledge in incremental fashion and accumulating conceptualization of new object categories. Apart from extensive training data for learning, a robot might always be confronted with an unknown objects and places in operating environment. Our framework handles this issue by processing visual information continuously and performs learning and recognition simultaneously. Our recognition model performs object detection and place recognition using convolutional neural network and semantic descriptor that is based on human perception system. The overview of our recognition model is described in Figure 4.

Our proposed recognition model consists of two stages: Training stage and Testing stag

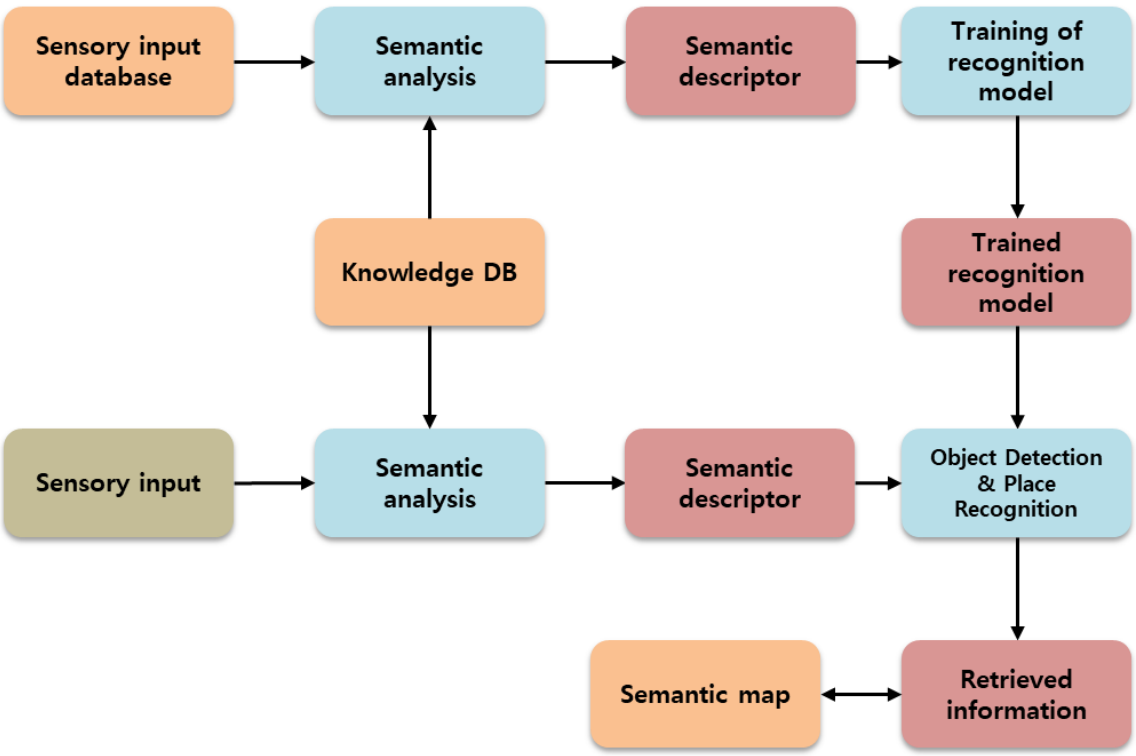


Figure 4: Semantic Descriptor-based Learning and Recognition Model

A. Training Stage

At training stage, we use CNN to train the object detection and place recognition model using our own indoor dataset for making the prediction using sensory input data and on-demand database. This stage is composed of three major components: semantic analysis, sematic descriptor and training the recognition model.

We perform semantic analysis for explicit and implicit models to get the semantic information and characteristics of each object. Two major operations related to preprocessing of visual data and feature extraction are involved in this step. We perform preprocessing to improve the performance of recognition model by reducing the noise from data for better local and global feature extraction and detection. We extract semantic object features from processed visual data including both global features and local features. We get the overall properties of each object by extracting the global features (edges, corners and color) while salient regions by retrieving the local features.

For semantic analysis, we extract the geometric features such as edges, lines, corners and shape in conjunction with metric information related to size and pose estimation of object are extracted and integrated into explicit model of our framework as global features. We store object properties and relationship between them as sensory input data while actions of an object such as movability as information of object's behavior in on-demand database.

The result of object analysis at semantic level is the extraction of semantic descriptions as per human perception. Thus, we reduce the semantic gap by combining the visual features extracted at low-level and information at high-level using semantic descriptor. We pass features vectors containing the geometric properties of the objects such as edges instead of the whole image to train our recognition model.

B. Testing Stage

At this stage, we run our recognition model in real world by performing the semantic analysis on the visual data and passing the feature vector to run our trained CNN model for object and recognition. Computational simplicity and minimum storage requirements are the major motivating factors for us to pass the extracted feature vectors instead of whole image to the recognition model. It also endows the robot with the ability of human-like perception and semantic understanding of the environment.

4 Experiment

We perform the real-world experiments in conventional center to evaluate the performance of our proposed semantic SLAM framework and extract the information of the environment and objects. These evaluations are conducted on an Intel Core i7-4712MQ 2.30 GHz CPU, NVIDIA GeForce 840M GPU, and 12GB RAM. Our recognition module uses ZED camera to detect objects and places while we perform localization and mapping using the data obtained from laser range finder (2D sensor).

We use TOSM to represent semantic information by establishing the concepts and linking the conceptual and physical objects of the environment. Figure 5. Shows the model of our environment in which operating areas is highlighted in red color.

Figure 5: Experiential Real-world Environment

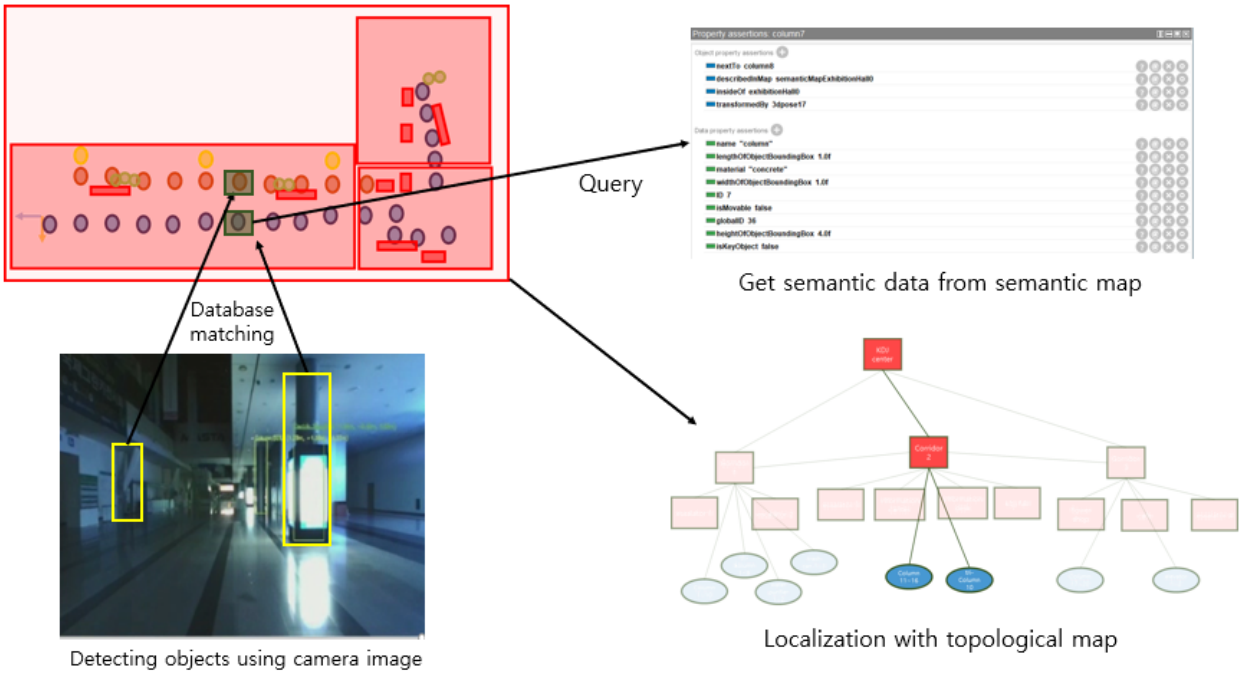


Figure 6: Experimental Result. (a) Recognized objects, (b) Semantic representation of Environment (c) Semantic database, (d) Topological map linking semantic environment with geometric information

Figure 6 shows all the steps involved in our real-world experiment. The robot localizes itself using topological map that endows the robot with spatial awareness. We build the semantic map by establishing semantic relationship between a place node in a topological map and its concepts. After that, we associate the objects that are recognized in a specific place with their topological nodes in the semantic map. Robot connects to the database that stores semantic information and properties of objects and places in order to match the relations. Figure 6(a) shows the objects recognized by our recognition model.

Our semantic map explains the structure of the environment at higher level that is closer to human-mapping system. Figure 6(b) shows the semantic representation of the environment, in which places are represented by rectangular boxes and objects are represented by circles. The blue circles indicate the columns, the orange circles show the vending machine along with the green circles that represent tri-columns while red boxes are places. Figure 6(c) shows the database that stores the ontology information for the robot mapping system and properties of the physical objects that are recognized when robot navigates in the environment. Our topological map shown in Figure 6(d) represents the environment by linking geometrical information and establishing the relations of semantic information to edges and nodes of relation graph. The proposed relation graph is focused on the environment mapping task and demonstrates semantic knowledge with the conceptual and spatial hierarchy. It represents the relationships between the information of corridor-1 containing elevators and columns along with the objects in coordidor-2 and coorid-3 that the robot knows.

We extract semantic map of our environment model based on occupancy grid as shown in Figure Figure 7 and add semantic concepts such as corridors and spatial relations like connectivity between different objects in the environment.

5 Conclusion

In our semantic SLAM framework, we have presented the central idea to endow the mobile robot with intelligent behavior. It has introduced the biological vision-based perception system for object and place recognition using CNN and semantic descriptor. Furthermore, we have proposed human brain inspired semantic mapping system to modulate the robot's behavior when it navigates in the environment to perform a task. Moreover, our TOSM approach represents the knowledge about the elements in the map. The experimental results indicate the feasibly of our prof framework in real-world indoor environment. In the future we plan to investigate to build

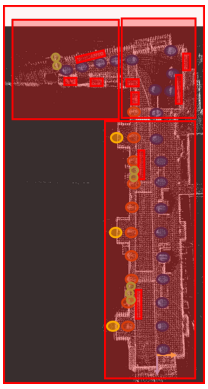


Figure 7: Semantic Map with Occupancy Grid Map

and updating a semantic map automatically without traditional maps and recognize objects and places using semantic map.

Acknowledgement

This research was supported by Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Ministry of Trade, Industry & Energy (MOTIE) (No. 1415162366 and No. 141562820)

References

[bohg17] J. Bohg et al., Interactive perception: Leveraging action in perception and perception in action *IEEE Trans. Robot* – Volume. 33, no. 6, pp. 1273–1291, 2017.

[Capobianco15] R. Capobianco, J. Serafin, J. Dichtl, G. Grisetti, L. Iocchi, and D. Nardi, A proposal for semantic map representation and evaluation, *2015 European Conference on Mobile Robots (ECMR)* 2015 - Proc., pp. 1—6, 2015.

[Pendleton17] S. Pendleton et al. "Perception, Planning, Control, and Coordination for Autonomous Vehicles," *Machines* – Volume. 5, no. 1, p. 6, 2017.

[Vasudevan08] Vasudevan, S. and Siegwart, R. Bayesian space conceptualization and place classification for semantic maps in mobile robotics *Robotics and Autonomous Systems* – Volume. 56, no. 6, pp. 522–537, 2008.

[Burgard07] W. Burgard, P. Jensfelt, R. Triebel, Ó . Martínez Supervised semantic labeling of places using information extracted from sensor data *Robotics and Autonomous Systems* – Volume. 55, no. 5, pp. 391–402, 2007.

[Nüchter08] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots *Robotics and Autonomous Systems* – Volume. 56, no. 11, pp. 915–926, 2008.

[Xie13] D. Xie, S. Todorovic, and S. C. Zhu. Inferring "dark matter" and "dark energy" from videos *IEEE International Conference on Computer Vision* – pp. 2224–2231, 2013

[cadena16] Cadena, Cesar, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age" IEEE Trans. Robot., *IEEE Transactions on robotics* – Volume. 32, no. 6, pp. 1309–1332, 2016

[Lowry16] S. Lowry et al. Visual Visual Place Recognition: A Survey *IEEE Transactions on Robotics* – Volume. 32, no. 1, pp. 1–19, 2016

[Sabourin10] C. Sabourin and K. Madani. Towards Human Inspired Semantic Slam *ICINCO 2010 - Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics* – Volume. 2, pp. 360–363, 2010

[Pronobis12] A. Pronobis and P. Jensfelt Large-scale semantic mapping and reasoning with heterogeneous modalities *IEEE International Conference on Robotics and Automation* – pp. 3515–3522, 2012.

[Bowman17] S. L. Bowman and G. J. Pappas. Probabilistic Data Association for Semantic SLAM *IEEE International Conference on Robotics and Automation (ICRA)* – pp. 1722–1729, 2017.

[Durand07] N. Durand et al Ontology-based object recognition for remote sensing image interpretation *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)* – Vol. 1, pp. 472–479, 2007.

[Ji12] Z. Ji et al. Towards automated task planning for service robots using semantic knowledge representation *IEEE 10th International Conference on Industrial Informatics* – pp. 1194–1201, 2012

[Riazuelo15] L. Riazuelo et al. RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach *IEEE Transactions on Automation Science and Engineering* – Volume. 12, no. 2, pp. 432–443, 2015

[Saigol15] Z. Saigol, M. Wang, B. Ridder, and D. M. Lane. The Benefits of Explicit Ontological Knowledge-Bases for Robotic Systems *Towards Autonomous Robotic Systems* – pp. 229–235, 2015.

[Azevedo18] H. Azevedo, J. P. Ribeiro Belo, and R. A. F. Romero, OntPercept: A Perception Ontology for Robotic Systems *Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)* – pp. 469–475, 2018.

[Lee18] S. Lee and I. Kim, A robotic context query-processing framework based on spatio-temporal context ontology *Sensors* – Volume. 18, no. 10, 2018.

[Sünderhauf18] N. Sünderhauf et al. The limits and potentials of deep learning for robotics *The International Journal of Robotics Research* – Volume. 37, no. 4–5, pp. 405–420, 2018.

[Zou19] Z. Zou, et al. Object Detection in 20 Years: A Survey *arXiv preprint arXiv:1905.05055* – pp. 1–39, 2019.

[Petrick08] R. Petrick et al. Representation and Integration: Combining Robot Control, High-Level Planning, and Action Learning *Proceedings of the 6th international cognitive robotics workshop* – pp. 32-41, 2008.

[Yan14] Yan, H., Ang, M.H. and Poo, A.N. A Survey on Perception Methods for Human-Robot Interaction in Social Robots *International Journal of Social Robotics* – Volume. 6, no. 1, pp. 85–119, 2014.

[Palomino16] Palomino, A.J., Marfil, R., Bandera, J.P. and Bandera, A. A new cognitive architecture for bidirectional loop closing *Robot 2015: Second Iberian Robotics Conference* – Volume. 418, no. November, pp. 721–732, 2016.

[Ye17] Ye, Chengxi, et al. What can i do around here? Deep functional scene understanding for cognitive robots *IEEE International Conference on Robotics and Automation (ICRA)* – pp. 4604–4611, 2017.

[Martinez17] Martinez-Martin, E. and Del Pobil, A.P. Object detection and recognition for assistive robots: Experimentation and implementation *EEE Robotics & Automation Magazine* – Volume. 24, no. 3, pp. 123–138, 2017.

[Ullah08] Ullah, Muhammad Muneeb, et al. Towards robust place recognition for robot localization *IEEE International Conference on Robotics and Automation* – pp. 530–537, 2008.

[Tasse16] Tasse, F.P. and Dodgson, N. Shape2vec: Semantic-based descriptors for 3D shapes, sketches and images *ACM Transactions on Graphics (TOG)* – Volume. 35, no. 6, pp.1–12, 2016.

[Zen12] Zen, Gloria, et al. Enhanced semantic descriptors for functional scene categorization *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* – pp. 1985–1988, 2012

[Panphattarasap18] Panphattarasap, P. and Calway, A. Automated Map Reading: Image Based Localisation in 2-D Maps Using Binary Semantic Descriptors *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* – pp. 6341–6348, 2018

The Semantics of Context: The Role of Interpretation and Belief in Visual Localization for Robots

Stephanie Lowry
Centre for Applied Autonomous Sensor Systems
Örebro University
70281 Örebro, Sweden
stephanie.lowry@oru.se

Abstract

While visual localization has improved in performance dramatically in recent years due to the development of ever-improving robust representations of locations, this paper considers a different aspect of the problem – the belief generation process. Belief generation is the conversion of a measure of similarity between two location representations to a measure of sameness – that is, are these two representations captured at the same location – which can be affected by the level of perceptual aliasing within the environment as well as the level of perceptual change. While probabilistic formulations of visual localization address these issues, environmental context can critically affect the performance of these belief generation methods.

1 Introduction

Visual localization is the process of using information about the external environment captured from vision-based sensors to correct a mobile autonomous system's belief about its location [LSN+16]. Visual localization systems have improved to an astonishing degree and modern systems are capable of impressive place recognition performance in extremely challenging conditions [SSD+15, AGT+18, KEC+18]. Much of this improvement is inspired by the rapid advances in machine learning for computer vision [LBH15] which allows localization systems to exploit sophisticated deep learning image representations. These robust representations can be used to compare two locations and determine if the appearance of these locations are similar.

A key challenge in visual localization is translating from the *similarity* of two locations to whether these two locations are actually the *same* place. Generally speaking, there are two scenarios that cause errors in visual localization: *perceptual aliasing* and *perceptual change*. These scenarios cause false positive and false negative errors respectively. Perceptual aliasing causes the system to believe observations from two separate locations are actually of the same place (false positive). Perceptual change causes observations of the same location not to be correctly identified as such (false negative).

Perceptual aliasing occurs when places in the environment look very similar and it is difficult to identify one location from another. Motorway driving can produce perceptual aliasing: motorways often look very similar over long distances, so you need to investigate highly distinctive elements – such as exit numbers on signs – to

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

know exactly where you are. Even if long stretches of the road appear superficially similar, you need to reject the possibility that they are actually the same place.

The second scenario is *perceptual change*: places in the environment do not look the same as they did on previous occasions. Perceptual change is particularly challenging when it happens uniformly over a region – for example, if day turns to night or snow falls – as all places in that region will become difficult to recognize. However, the spatial relationship between places does not change, so as long some similarity in appearance remains, a weak location hypothesis can be formed and by observing multiple nearby locations and the spatial relationship between them, a system can gradually build up confidence in its location belief.

There is an inherent conflict between resolving perceptual aliasing and perceptual change: perceptual aliasing requires adhering to a strict matching strategy where places must be both highly similar and highly distinctive, while perceptual change requires a permissive matching strategy where places may be matched together even when they do not appear similar at all. Thus an important consideration for a localization system is *context* – is the system in a situation when it should demand highly rigorous matching expectations or is a more permissive strategy necessary?

2 Belief Generation

As discussed in [LSN+16], there are a number of methods for determining whether two location representations were captured at the same location, such as voting methods [SBS07] and – when techniques inspired by text-based document analysis were used – the term frequency–inverse document frequency (TF-IDF) was used [NG12] to measure the mutual information between representations. A probabilistic formulation was also often used. Using a probabilistic framework has advantages: it provides a mechanism for managing uncertainty introduced from various sources and it naturally outputs a measure that expresses the degree of confidence in the current location belief. One well-known probabilistic framework for visual localization is FAB-MAP [CN08]. FAB-MAP uses a Naïve Bayes or a Chow-Liu [CL68] approximation to simplify the complex joint probability between the visual words in its model, and introduces a hidden variable to reduce the probabilities to quantities that can be calculated from training data.

An important aspect of location matching is geometric consistency – elements within the environment should stay in the same physical position relative to each other. Geometric verification tests can eliminate false positive matches using spectral clustering [Ols09] or RANSAC [FB81]. Furthermore, not only will elements within a location remain geometrically consistent, but the spatial relationship of the locations themselves will remain constant. These spatial relationships can be integrated into the localization belief probabilistically [BHK12] or via other methods such as network flows [NSBS14]. The spatial relationship between locations is extremely important information – in an extreme case, impressive localization results can still be achieved when appearance information is ignored and only odometry information is used [BGU16].

A number of trade-offs between competing priorities have been observed. A localization system becomes increasingly dependent on spatial information when there is extreme perceptual change [MW12, NSBS14, HB14], thus increasing sensitivity to motion uncertainty and reducing viewpoint flexibility. It has also been shown that methods that perform more effectively in perceptually changing environments do not provide as accurate localization [SMT+18]. These trade-offs suggest a choice must be made as to the requirements of the application and the operating environment.

3 Probabilistic Belief Generation

This section presents the formalism behind a probabilistic formulation of visual localization. Using a probabilistic framework has many advantages: it provides a mechanism for managing uncertainty introduced from various sources and naturally outputs a measure that expresses the degree of confidence in the current location belief. Probabilistic localization also naturally integrates some spatial environmental context; if there is a great deal of perceptual aliasing in the environment, the system's confidence in its location belief will be low.

The probabilistic framework is applied recursively over time as the system moves through the world: the prior belief is updated based on the system's motion model. The uncertainty in the system's location belief is continuously increased by the error in the motion model, and would grow in an unbounded manner if it were not constrained and corrected by the external observations of the world.

Formally, visual localization is probabilistically defined as follows (using the same notation as [CN08]): at time step k, the system has made a series of observations $\mathcal{Z}^k = \{Z_0, Z_1, \ldots, Z_k\}$, and has previously visited locations

$\{L_0, L_1, \ldots, L_{k-1}\}$. The likelihood of the system being in location L_i at time k given the current observation Z_k is

$$p(L_i \mid \mathcal{Z}^k) = \frac{p(Z_k \mid L_i, \mathcal{Z}^{k-1}) p(L_i \mid \mathcal{Z}^{k-1})}{p(Z_k \mid \mathcal{Z}^{k-1})}. \tag{1}$$

The second term in the numerator $p(L_i \mid \mathcal{Z}^{k-1})$ is the ***location prior***: the system's prior belief about the location before making the current observation Z_k. It allows the localization to be updated recursively over multiple timesteps: at time step $k+1$ the output of Equation 1 becomes the new location prior and Equation 1 can be applied again.

The other two terms are the ***observation likelihood models***. The first term in the numerator $p(Z_k \mid L_i, \mathcal{Z}^{k-1})$ is the likelihood that the robot would make observation Z_k if it is indeed at location L_i, and the denominator $p(Z_k \mid \mathcal{Z}^{k-1})$ is a normalizing factor that determines the likelihood that the robot would make the observation Z_k anywhere within the environment, thereby introducing further spatial context into the localization calculation.

4 Observation Likelihood Models

The observation likelihood models are the mechanisms by which context is introduced into the localization calculation. The system's observations naturally have a degree of uncertainty and have to be interpreted within the context of the environment: for example, should the system have a strict or a permissive matching strategy? Furthermore, the probabilistic framework does not implicitly ensure temporal environmental context (that is, perceptual change) is included.

The observation likelihood models themselves need to be learned or at least embody some data-driven assumptions about the environment. For example, FAB-MAP learns its likelihood models from training data [CN08]. Many visual localization systems employ learning techniques to improve the performance of its location representations, including using state-of-the-art deep learning to learn about appearance change [LGMR18] or viewpoint change [GSM18], and learning about the observation likelihoods is closely related to the chosen image representation.

The performance of the observation likelihood model can be assessed independently of the performance of the image representations. The likelihood models depend on prior belief about the likelihood of appearances which naturally vary both due to the environment itself and to the conditions under which the environment is observed.

The performance of a system can depend critically on the correctness of the likelihood model. In [Low14], a probabilistic visual localization system was evaluated on the St Lucia dataset [GMMW10]. Two observation likelihood models ($M1$ and $M2$) were trained on the data – $M1$ was trained using the data from the same time of day as the test data and $M2$ was trained using data from a different time of day. The performance of the system increased from correctly localizing in 10% of places using $M1$ to correctly localizing in over 70% of places using $M2$, with all other aspects of the system remaining unchanged.

Since the data used to train $M1$ was only captured a few hours away from the data used to train $M2$, these results suggest that not only is the correct context necessary for generating a correct location belief, but that the system must be flexible to change as the appearance of the environment varies. In fact, in some circumstances a dynamically generated likelihood model approximated online using current environment data can out-perform a pre-trained likelihood model that was calculated using exact ground truth data captured only a few hours earlier [Low14]. However, a model that is approximated online also contains assumptions about the environmental context [LM15]. If these assumptions are incorrect, it can also negatively affect the localization performance.

5 Conclusion

Visual localization has made transformative progress in recent years, with existing image description methods able to perform robustly in impressively challenging scenarios of perceptual change and differing viewpoints. These robust image description methods can be used to evaluate similarity between different location representations, which a belief generation system can convert into a likelihood or confidence metric. However, belief generation methods are sensitive to the environmental context in which the system is operating. Thus training of the models for a belief generation system must be appropriate for the current environmental conditions, and if a system is operating in a dynamic, perceptually varying environment the belief generation models must reflect this variation.

Acknowledgement

This work was supported by the Swedish Research Council (grant no. 2018-03807).

References

[AGT+18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, June 2018.

[BGU16] M. Brubaker, A. Geiger, and R. Urtasun. Map-based probabilistic visual self-localization. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2016.

[BHK12] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1635–1642, May 2012.

[CL68] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, May 1968.

[CN08] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[FB81] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[GMMW10] A. Glover, W. Maddern, M. Milford, and G. Wyeth. FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3507–3512, May 2010.

[GSM18] S. Garg, N. Sünderhauf, and M. Milford. LoST? appearance-invariant place recognition for opposite viewpoints using visual semantics. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[HB14] P. Hansen and B. Browning. Visual place recognition using HMM sequence matching. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4549–4555, Sept 2014.

[KEC+18] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier. A Holistic Visual Place Recognition Approach using Lightweight CNNs for Severe ViewPoint and Appearance Changes. *arXiv e-prints*, Nov 2018.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, May 2015.

[LGMR18] Y. Latif, R. Garg, M. Milford, and I. Reid. Addressing challenging place recognition tasks using generative adversarial networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2349–2355, May 2018.

[LM15] S. Lowry and M. Milford. Building beliefs: Unsupervised generation of observation likelihoods for probabilistic localization in changing environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3071–3078, Sept 2015.

[Low14] S. Lowry. *Visual place recognition for persistent robot navigation in changing environments*. PhD thesis, Queensland University of Technology, 2014.

[LSN+16] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, Feb 2016.

[MW12] M. Milford and G. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1643–1649, May 2012.

[NG12] T. Nicosevici and R. Garcia. Automatic visual bag-of-words for online robot navigation and mapping. *IEEE Transactions on Robotics*, 28(4):886–898, Aug 2012.

[NSBS14] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2014.

[Ols09] E. Olson. Recognizing places using spectrally clustered local matches. *Robotics and Autonomous Systems*, 57(12):1157–1172, 2009.

[SBS07] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. pages 1–7, June 2007.

[SMT+18] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[SSD+15] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of ConvNet features for place recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, Sept 2015.

Dense-Loop: A Loop Closure Detection Method for Visual SLAM using DenseNet Features

Chao Yu
Tsinghua University, Beijing, China 100084
yc19@mails.tsinghua.edu.cn

ZuXin Liu
Beihang University, Beijing, China, 100191
xinye@buaa.edu.cn

Xin-Jun Liu*, Fei Qiao*, Yu Wang, Fugui Xie, Qi Wei,Yi Yang
Tsinghua University, Beijing, China 100084
{xinjunliu, qiaofei, yu-wang, xiefg, weiqi, yangyy}@mail.tsinghua.edu.cn

Abstract

Loop closure detection (LCD) is an important part in SLAM for the autonomous mobile robot. A recent trend is to employ off-the-shelf networks' features to address LCD problem, which outperform traditional hand-crafted features. However, what kind of network is more suitable in LCD and how to use their CNN features have not been well-studied. In this paper, we compare many popular networks and introduce DenseNet in this field. The features extracted by DenseNet, which preserve both semantic information and structure details, outweigh other popular CNN features signicantly. Then a DenseNet feature-based framework (Dense-Loop) is proposed to address the LCD problem. We use the Weighted Vector of Locally Aggregated Descriptor (WVLAD) method to encode the local descriptors as the final global descriptor, which could resist geometry structure and viewpoint changes. Furthermore, 4 max-pooling by channel and locality-sensitive hashing (LSH) are adopted to ensure the real-time search. Extensive experiments are conducted on public datasets using Precision-Recall Curve evaluation method. The results demonstrate Dense-Loop could achieve state-of-the-art performance.

1 Introduction

In recent years, the combination of semantics and SLAM has become a research hotspot, and many related works have appeared, such as DS-SLAM[YLL+18], DA-RNN[XF17] and so on. Most of these SLAM systems utilize semantics in Visual Odometry (VO) and Mapping, while introducing semantic information into loop closure detection (LCD) is indispensable and requires further research.

Visual place recognition is a basic part in re-localization and loop closure detection for mobile robots[LSN+16]. If the robot could determine whether an image of a place has been visited before, then this information could help the robot re-localize itself, or correct the error and drift accumulated in the simultaneous localization and mapping (SLAM) process[LM13, MAT17].

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

However, this problem is very challenging. On the one hand, the same place may have different appearances at different time due to the illumination or viewpoint changes. On the other hand, two different places may have the similar texture and appearance. A false positive recognition of a place may corrupt the global optimization process and cause severer unrecoverable localization and mapping failure[Cum08].

Many effective methods have been proposed to solve loop closure detection problem in robotics field. One of the most prevalent methods is visual bag-of-words (BoWs)[MAT17, Cum08], which treats descriptors of local features as visual words. This kind of method can achieve good performance on place recognition, and it is robust against viewpoint changes. However, the hand-crafted features can hardly deal with environment changes, such as the illumination changes and similar textured regions[Cum08, UMCM14, GSM18].

Recently, many researchers have found the features extracted from off-the-shelf convolutional neural networks (CNN) have better performance than hand-crafted features[KSH12] and began to investigate how to use CNN features in LCD[CLJM14, SSD+15, AGT+18, SSJ+15, BWZ+16]. Even so, the research in this field is preliminary and incomplete, partially because of the weak interpretability of neural networks.

Before delving into the paper, we first see some frequently asked questions when people want to employ CNN in LCD. First of all, there are numerous outstanding neural network architectures, which one is more suitable for LCD and what is the reason? Secondly, CNN features vary from hand-crafted features in respect of the quantity and dimension. Is traditional loop closure detection framework (such as BoWs) suitable for CNN features? If not, do we have better solutions?

In this paper, we try to explore these problems in depth and give corresponding explanations. The main contributions include:

1. We compare many off-the-shelf networks and find DenseNet outweighs other popular networks in loop closure detection, because this dense-connected network could preserve both semantic information and structure details of the input image.

2. A loop closure detection framework (Dense-Loop) using DenseNet features is proposed in this paper. Decoupling by feature-maps (DBF) and Weighted Vector of Locally Aggregated Descriptor (WVLAD) method is utilized to make full use of DenseNet features according to its own distinctions.

3. Extensive experimental results show Dense-Loop approach could achieve state-of-the-art performance on public datasets.

In the rest of the paper, the structure is as follows. Section 2 briefly introduces some current accomplishments of loop closure detection. Section 3 presents the proposed framework in detail. Subsequently, extensive comparative experiments and evaluation are presented in Section 4. Finally, a brief conclusion and the future work are summarized in Section 5.

2 Related works

We categorize current accomplishments on loop closure detection into three groups: traditional hand-crafted feature-based approaches, end-to-end training approaches, and approaches based on the CNN features extracted from off-the-shelf networks.

Many well-designed local features are widely used in place recognition and loop closure detection tasks because their ability to resist scale changes or orientation changes. One of the most successful use is FAB-MAP, which employs SUFT[BETG08] and BoWs for place recognition and demonstrates robust performance against viewpoint changes[Cum08].[MAT17] integrate ORB[RRKB11] and BoWs in SLAM. This kind of method becomes the most popular framework to detect loop closure in real-time visual SLAM systems. However, these hand-crafted features only care about low-level information of the image and can hardly deal with environment changes, such as illumination changes. Furthermore, these statistics based methods' performance depends heavily on the quality of the features and may be easily deceived by the textured dynamic objects in the environment.

Considering the shortcomings of the hand-crafted features, a recent trend in loop closure detection is to train a CNN network in an end-to-end manner. NetVLAD[AGT+18] is a novel architecture which aims to minimize the distance of two image representations of the same place. The training images are categorized into many tuples, where each training query image has corresponding potential positive samples and definite negative samples.[LAGOPGJ17] adopt the similar triplet training scheme and could produce a 128 dimension descriptor vector for each image. However, all of these supervised learning approaches require a large amount of labeled datasets to train. It is also a bottleneck for others to use the network for their own needs.

Another trend is to exploit the learned features of the off-the-shelf networks with pre-trained weights. [CLJM14] employ CNN features based on OverFeat for place recognition. The performance of feature-maps

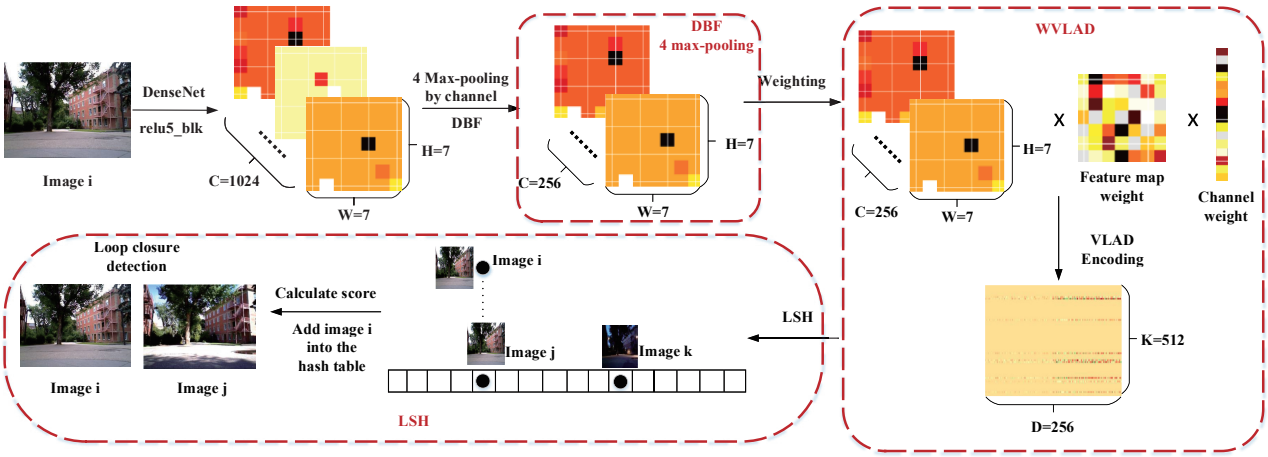


Figure 1: The pipeline of Dense-Loop

of different layers is explored. [HZZ15] focus on using AlexNet to generate an image representation appropriate for visual loop closure detection in SLAM. They find CNN features outperform hand-crafted features when illumination changes signicantly.[SSD+15] deploy pre-trained AlexNet as CNN features and using locality-sensitive hashing and semantic search space partitioning optimization techniques to ensure real-time search. These kind of methods do not require specific end-to-end training and thus are more convenient. The feature could be extracted without interference to the pre-trained networks that designed for other tasks. However, since there are numerous outstanding network architectures in recent years, which one is better and how to make good use of its inner features have not been fully explored.

In this paper, we will explore what kind of network is more suitable in LCD and how to use them to achieve better performance without specific supervised training.

3 Framework of Dense-Loop

In the proposed framework, the output of ReLu layer in the last dense block of DenseNet is adopted as the initial features and decoupling by feature-maps (DBF) is utilized to decompose the global feature into local descriptors. Then, 4 max-pooling by channel is adopted to reduce the computational complexity. Finally, Weighted Vector of Locally Aggregated Descriptor (WVLAD) method is proposed to improve the ability of resisting scale or viewpoint changes. To accelerate the searching process, locality-sensitive hashing (LSH)[RPH05] is employed according to the characteristic of Dense-Loop descriptors. The pipeline of Dense-Loop is shown in Figure 1, where C, H, W represent the dimension of the channel, the height and weight of feature-maps. K is the number of cluster centers and D represents the dimension of one cluster center.

3.1 Image descriptors extraction

In the traditional BoWs, a lot of disordered local descriptors with low dimensions are extracted and they are designed to resist scale or viewpoint changes. However, CNN features are ordered and 3-dimension. Therefore, the first thing is to exact good features from CNN and map them to 2-dimension.

3.1.1 DenseNet features

DenseNet is a compact network and made up of dense blocks. All layers in one dense block are directly connected to ensure maximum information flow between feature-maps. The input of each layer is all the preceding layers' output, and thus, the block's final classifier could obtain all the information of the previous feature-maps. This kind of compact internal representation could reduce feature redundancy and help to solve vanishing-gradient problem. The architecture of a 5-layer block in DenseNet is shown in Figure 2(a). DenseNet adopted in Dense-Loop is made up of 5 dense blocks. The output of ReLu layer in the last dense block is used as the raw features of the input image, where 7×7 is the size of feature-maps and 1024 is the number of channels. The reason for choosing the ReLu layer is that it is cleaner and contains less noise.

The reason of using DenseNet is its reuse of feature-maps. The features of low layers contain more structural information and measure fine-grained similarity, which is similar to hand-crafted features. While the features of higher layers care more about semantic information and measure semantic similarity. A natural idea is to utilize the complementary of high-layer and low-layer features. The outputs of last few layers preserve all extracted features of preceding layers, which means, the low-level features and high-level features are merged together in an efficient way. It is helpful for more fine-grained features expression of an image. The superiority of DenseNet will be illustrated in the experiment section in detail.

3.1.2 DBF and 4 max-pooling by channel

Here are two ways to map these features to 2-dimension, as shown in Figure 2(b). One is decomposing the global feature into 49 local descriptors with 1024 dimensions, called decoupling by feature-maps (DBF). Anther way is to decompose 1024 local descriptors with 49 dimensions, called decoupling by channel (DBC). The former plan is chosen because it is of physical meaning, and it has better performance than DBC. Each pixel in the feature-map is corresponding to a receptive field in the input image, and all the channels of the pixel could describe the distinctions of the corresponding receptive field. As for DBC, it's more like using many global descriptors to describe an image. But image's viewpoint change may cause a shift in the feature-maps and thus the ability to resist geometry structure or viewpoint changes will be weaken.

In order to ensure the real-time search, a method called 4 max-pooling by channel is proposed to reduce the descriptors' dimensions with minimal accuracy reduction. 1024-dimension descriptors are divided into 256 groups and the maximum value of each group is used as the final descriptor. Compared with PCA, which is widely used to reduce dimensions, 4 max-pooling by channel has less computational complexity but similar performance. More results can be found in the experimental part.

3.2 WVLAD method

In the traditional BoWs, BoW encoding method is used to measure the similarity of two images. BoWs is a statistical method and usually needs a large number of visual words (e.g. 10^6) in the dictionary. A lot of local descriptors with low dimensions are more suitable in this situation, while the CNN descriptors, which are decoupled by feature-maps, often have small quantity but large dimensions. Besides, it is hard to train such a huge BoW dictionary. Instead, Weighted Vector of Locally Aggregated Descriptor (WVLAD) is proposed in this paper to encode the 49×256 local descriptors of an image.

WVLAD could ignore the geometric structure of the image via clustering and care more about the distinctions via weight. Therefore, it's more resistant to viewpoint and scale changes than calculating euclidean distance of CNN features. It is an improved method of famous Vector of Locally Aggregated Descriptor (VLAD)[JDSP10] method and inspired by Cross-dimensional Weighting for Aggregated Deep Convolutional Features (CROW)[KMO16] method.

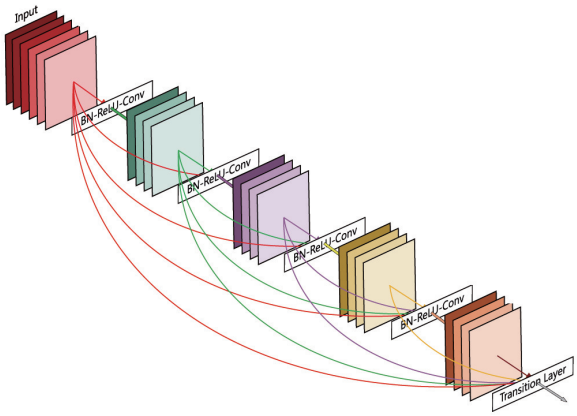
Usually we want the descriptors care more about the distinctions of the image and reduce the importance of the plain areas (e.g. sky). It's similar to the human perception system, which is conducive to improving resistance to environment changes. One way is to use region proposal methods and compute regions' descriptors respectively. Another way is to adopt the self-adaptive weight methods to adjust the importance of the textured regions and ordinary areas. The first way is computational expensive. Considering the need for real time, the second way is integrated in Dense-Loop. Figure 3 shows the detailed process of calculating the feature-maps weight (FW) and the channel weight (CW).

The strong response of convolution is usually corrsponding to the region of objects. FW can force features to focus on the textured regions and help solving scale changes. Let $F \in \mathbb{R}^{(C \times H \times W)}$ denotes the 3-dimension features of the inner layer. $X \in \mathbb{R}^{(H \times W)}$ represents one feature-map. c, h, w is the location of the feature vector. $FW \in \mathbb{R}^{(H \times W)}$ can be calculated by summing feature-maps of all channels. Then L2-norm and a power normalization with power 0.5 are utilized to get aggregated feature-maps weight.

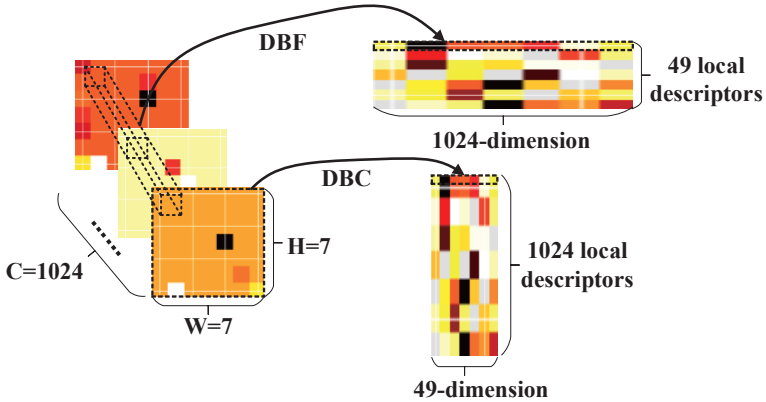
$$S = \sum_c X_c \tag{1}$$

$$S' = \sqrt{\sum_{h,w} S_{h,w}^2} \tag{2}$$

$$FW = \sqrt{S/S'} \tag{3}$$



(a) A 5-layer dense block



(b) DBF and DBC

Figure 2: (a) A 5-layer dense block with a growth rate of k = 4. The figure is reproduced from[HLvMW17]. (b) The description of DBF and DBC.

$CW \in \mathbb{R}^{(1\times C)}$ is similar to the idea of inverse documentary frequency (IDF) in BoWs, that is, reducing the importance of high-frequent features.

$$T_c = \frac{\sum_{X_{h,w}>0} 1}{H \times W} \tag{4}$$

$$CW_c = \begin{cases} \log(\frac{\sum_{c=1}^{C} T_c}{T_c}), T_c > 0 \\ 0, T_c = 0 \end{cases} \tag{5}$$

Then, we can calculate the weighted feature-maps $F_{weight} \in \mathbb{R}^{(C\times H\times W)}$. And decompose it into weighted local descriptors L, which means 49 local features with 256 dimensions.

$$F'_c = F_c \times FW \tag{6}$$

$$F_{weight} = F'_{c,h,w} \times CW_c \tag{7}$$

In order to improve the ability of resisting geometry structure or viewpoint changes, VLAD is used to encode weighted local descriptors as a global descriptor. Firstly K-means is used to cluster all the weighted local descriptors of the datasets and get the codebook $\{u_1, ..., u_K\}$, where K is the number of cluster centers. Each local descriptor L_i has its corresponding cluster center u_j: $NN(L_i) = argmin_j \|L_i - u_j\|$, where NN represents nearest neighbor. VLAD is denoted as a set of vector $V = [v_1^T, ..., v_K^T]$, where each v_i is associated with a cluster center u_i and has the same size. Then V is calculated by the concatenation of the residual of each L_i and $NN(L_i)$:

$$v_i = \sum_{L_t:NN(L_t)=i} L_t - u_i \tag{8}$$

Finally, a power normalization with power 0.5 and L2-norm is utilized to normalize V.

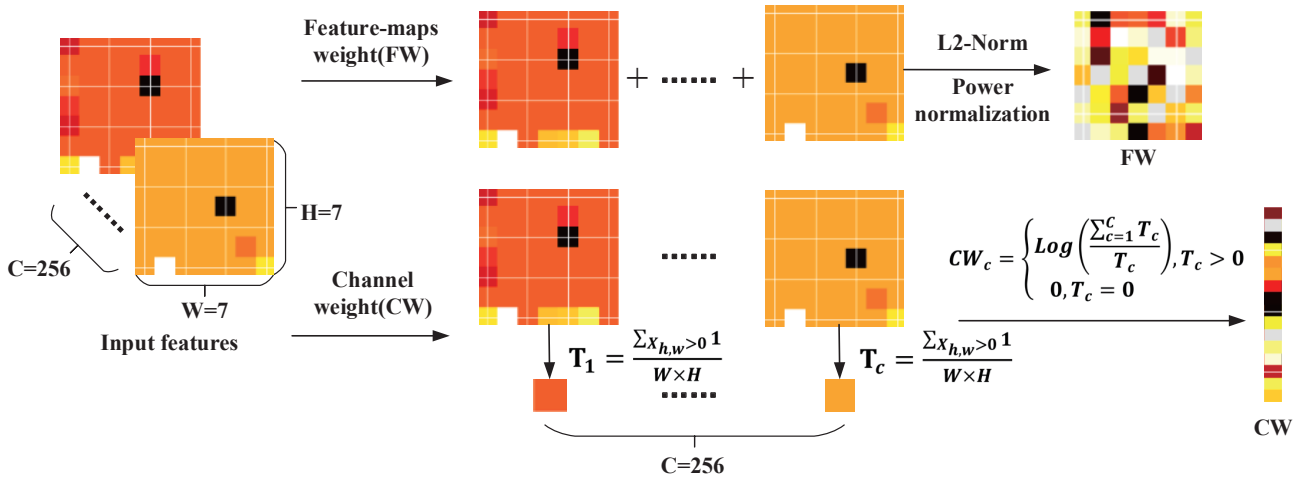


Figure 3: The detailed process of calculating FW and CW.

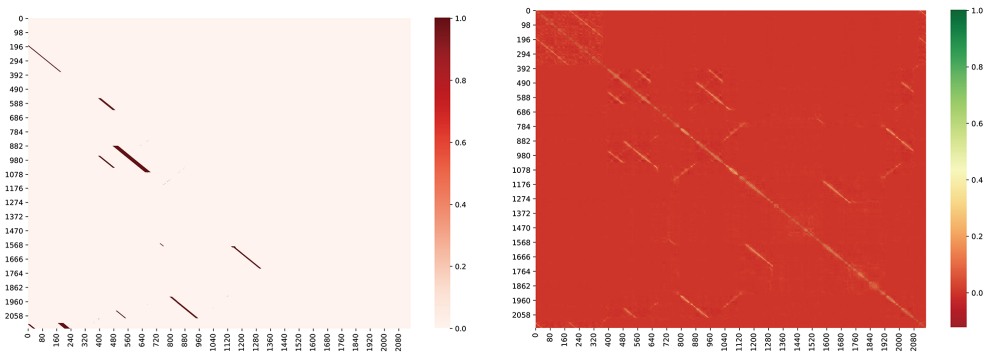
3.3 Locality-Sensitive Hashing

An important feature of loop closure detection for robotic application (e.g. SLAM) is real-time. In the traditional BoWs, K-D tree is adopted as the nearest neighbor search. However, the spatial dimension of Dense-Loop descriptors is far more than the number of words in the codebook, K-D tree will be unsuitable in such case. Instead, locality-sensitive hashing (LSH) is employed to speed-up the search with minimal accuracy degradation. The detailed process is shown in Figure 1. The Hamming distance between the respective hashed bit vectors, which is a cheap operation, is used to evaluate the similarity. According to our test, using 1024 bits retains approximately 99% performance but much more quick than brute search.

4 Experimental Results and Explanations

4.1 Datasets and evaluation method

City Center dataset[Cum08] and New College dataset[Cum08] are widely used in visual SLAM research and loop closure detection evaluation in particular. The former dataset has many dynamic objects like pedestrians and vehicles. Besides, the sunlight, wind and viewpoint change may cause the features like shadow unstable. The latter New College dataset has many dynamic elements and repeated elements, such as similar walls and bushes. Ground truth are given in two datasets. Figure 4 shows the ground truth and the results of Dense-Loop.



(a) Ground truth (b) Results of Dense-Loop

Figure 4: The ground truth and the results of Dense-Loop on New College Dataset. Pixel (i, j) represents the relationships of image i and image j.

However, the provided ground truth can't be used directly. It's inconsistent with the goal of loop closure detection because we only need to identify one loop in the same place. Therefore, new definition of the true loop are made based on the original ground truth. The images in one dataset are divided into two groups, named left

and right, and so is the ground truth. If a loop is detected, we will stop searching loop in 10 images (according to GPS) to avoid getting the same loop. When we vary the threshold if a loop closure is accepted, the precision and recall value will change and the PR-Curve can be gained.

4.2 Experiments and evaluation

Some comparative experiments are conducted to explore the validity of Dense-Loop. Dense-Loop could achieve state-of-the-art performance on public datasets, The reason can be summarized as two points. One is excellent features from DenseNet, which take high-level semantic information and fine-grained information into account. Another is WVLAD method, which could ignore the geometric structure of the image via clustering and care more about the distinctions via weight.

4.2.1 Why DenseNet

In recent years, there are many prevalent and excellent convolutional networks showing up, such as ResNet50[HZRS16], VGG[SZ14], DPN[CLX+17], SENet[HSS17], ResNeXt[XGD+17], NasNet[ZVSL17], SqueezeNet[IMA+16], Xception[Cho17], Inceptionv3[SVI+15], Inceptionv4 and Inception-ResNet[SIV17]. To verify the excellent features of DenseNet, extensive comparative experiments were conducted. Figure 5 exhibits the PR-Curves of different networks on New College dataset. Curves are named by the following formats: network name_layer name. For example, DenseNet_relu5_blk represents the features extracted from relu5_blk layer of DenseNet. All the networks are pre-trained on the ImageNet2012 dataset and euclidean distance is adopted as the similarity score. The layer with best performance in each network is chosen to draw in the figure and it is apparent that DenseNet outweighs other popular network architectures.

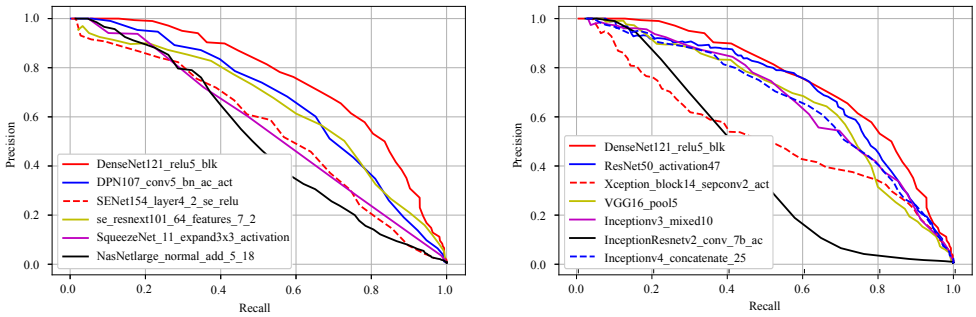
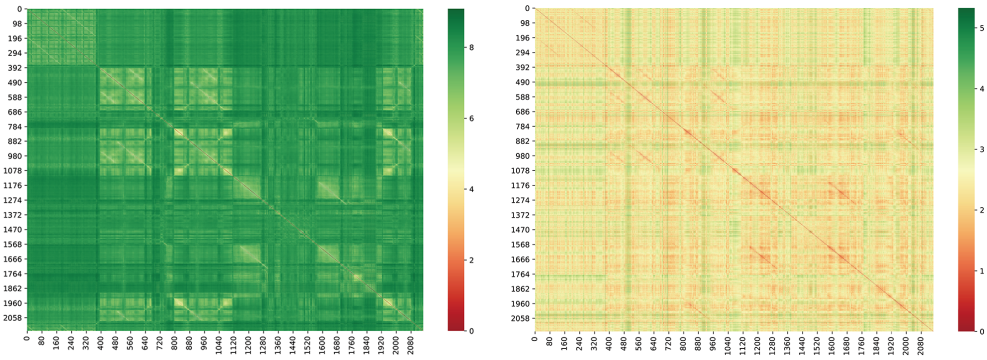


Figure 5: The PR-Curves of different networks on New College dataset.

Figure 6 shows the euclidean distance of images on New College dataset when employing DenseNet and Xception respectively. The high-level features of Xception, which care more about semantic information, have a poorer discrimination on images than those of DenseNet. A common method to combine various levels' features is to concatenate them directly, but DenseNet aleady did this during the forward processing. The output of the last few layers integrate both low-level and high-level features naturally.

4.2.2 Why DBF and 4 max-pooling

Figure 7(a) shows the PR-Curves of DBF and DBC on City Center dataset. In order to make a quick comparison, euclidean distance is adopted as the similarity score. It is obvious that DBF far outweighs DBC and similar results can be gained on New College dataset. Figure 7(b) illustrates the PR-Curves of different dimensionality reduction methods on City Center dataset. The label named relu5_blk means the original features without dimensionality reduction. The label named 4 max-pooling by channel represents applying 4 max-pooling to the feature's channel dimension. The label named 256 PCA means reducing the channel dimension to 256 through PCA method. We can observe that utilizing 4 max-pooling by channel can maintain 99% accuracy and have alomost the same performance as PCA. Considering the processing time, 4 max-pooling by channel is adopted finally.



(a) DenseNet (b) Xception

Figure 6: The euclidean distance of images on New College Dataset.

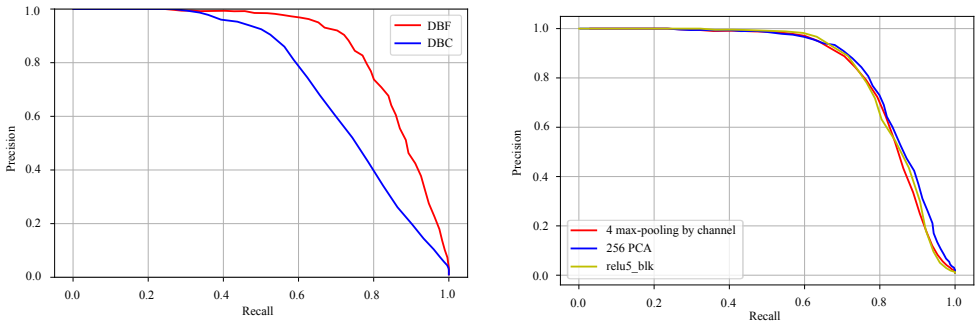


Figure 7: The PR-Curves of DBF V.S. DBC and different dimensionality reduction methods on City Center dataset.

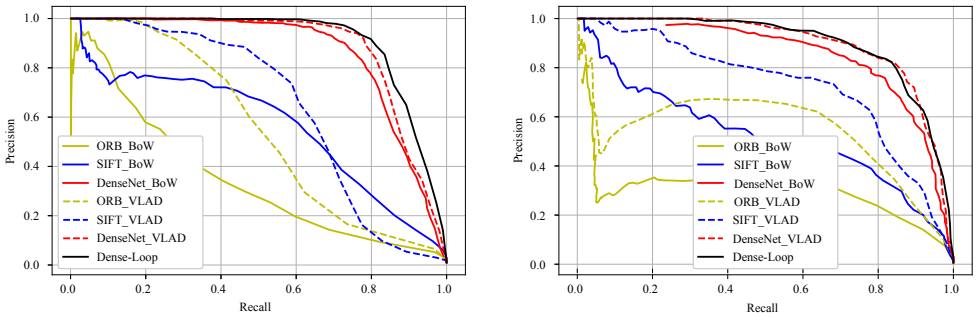
4.2.3 Why WVLAD

In order to compare the performance with traditional methods, two hand-crafted features (ORB and SIFT) and two encoding methods (BoW and VLAD) are adopted. The VLAD codebooks have 512 cluster centers, just the same as Dense-Loop, while BoW codebooks have 10000 visual words. The results on two datasets are shown in Figure 8.

It's clear that WVLAD could achieve better performance than BoW and VLAD encoding method based on DenseNet. And we can notice Dense-Loop far outweighs hand-crafted features. Here are two typical examples. In Figure 9(a) and 9(b), high similarity score is obtained based on hand-crafted features because of similar textured regions on the trees and sky, while score of Dense-Loop is close to zero in this case. This is because Dense-Loop could utilize high-level semantic and global information to judge the similarity. In Figure 9(c) and 9(d), Dense-Loop can recognize the two images as the same place with high score but hand-crafted features can't achieve that due to illumination changes. Besides, in this case, we can also find Dense-Loop can resist the viewpoint changes. As for WVLAD and VLAD, WVLAD can reduce channel redundancy by CW and focus on the distinguished and unique parts of the image by FW. Therefore, better performance can be obtained in some cases by solving the problem of scale and viewpoint changes.

5 Conclusion

Loop closure detection is used to detect if the robot has passed through the same place. It's crucial for the robot to establish a globally consistent map, especially for large and long-term scenes. A framework of loop closure detection based on CNN features is proposed in this paper. We find that features extracted from DenseNet outweigh hand-crafted features and other popular networks' features. The reason is DenseNet can preserve both semantic information and structure details of the input image via dense connection. In order to improve the ability of resisting scale or viewpoint changes, decoupling by feature-maps (DBF) and Weighted Vector of Locally Aggregated Descriptor (WVLAD) method is utilized to make full use of DenseNet features according to its own distinctions. Locality-sensitive hashing (LSH) and 4 max-pooling by channel are adopted to ensure the



(a) City Center dataset (b) New College dataset

Figure 8: The PR-Curves of DenseNet V.S. (ORB, SIFT) and Dense-Loop V.S. (BoW, VLAD)

(a) (b) (c) (d)

Figure 9: Picture (a) and (b) with ORB features come from different scenes, but they share similar textured regions (e.g. trees and sky). Picture (c) and (d) with ORB features come from the same place, but they have different apparences, such as illumination changes.

real-time search for robotic application. Extensive experiments illustrate Dense-Loop approach could achieve state-of-the-art performance on public datasets.

However, the impact of the training datasets on the network's performance has not been investigated. In the future, we will conduct more extensive experiments to explore the generalization ability of Dense-Loop, which is important in real-world robot applications. Besides, we would consider to utilize semantic information of the network's prediction results and establish a multi-level semantic knowledge base to speed up the search and improve the loop closure detection performance.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 91648116 and 51425501.

References

[AGT+18] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, June 2018.

[BETG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BWZ+16] Dongdong Bai, Chaoqun Wang, Bo Zhang, Xiaodong Yi, and Yuhua Tang. Matching-range-constrained real-time loop closure detection with cnns features. *Robotics and Biomimetics*, 3(1):15, Sep 2016.

[Cho17] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 1800–1807, July 2017.

[CLJM14] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509, 2014.

[CLX+17] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. *CoRR*, abs/1707.01629, 2017.

[Cum08] M Cummins. Fab-map : Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[GSM18] S. Garg, N. Suenderhauf, and M. Milford. Don't look back: Robustifying place categorization for viewpoint- and condition-invariant place recognition. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3645–3652, May 2018.

[HLvMW17] G. Huang, Z. Liu, L. v. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, July 2017.

[HSS17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[HZZ15] Y. Hou, H. Zhang, and S. Zhou. Convolutional neural network-based image representation for visual loop closure detection. In *2015 IEEE International Conference on Information and Automation (ICInfA)*, pages 2238–2245, Aug 2015.

[IMA+16] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[JDSP10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, June 2010.

[KMO16] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 685–701, Cham, 2016. Springer International Publishing.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[LAGOPGJ17] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92:89–95, 2017.

[LM13] Mathieu Labbe and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, June 2013.

[LSN+16] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David Cox, Peter Corke, and Michael J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[MAT17] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[RPH05] Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. Randomized algorithms and nlp: Using locality sensitive hash function for high speed noun clustering. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 622–629, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

[SIV17] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Multi-scale orderless pooling of deep convolutional activation features. In *Proceeding of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pages 4278–4284, 2017.

[SSD+15] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, Sept 2015.

[SSJ+15] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems (RSS)*, Auditorium Antonianum, Rome, July 2015.

[SVI+15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[UMCM14] B Upcroft, C Mcmanus, W Churchill, and W Maddern. Lighting invariant urban street classification. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1712–1718, Hong Kong, China, 2014. IEEE.

[XF17] Yu Xiang and Dieter Fox. DA-RNN: semantic mapping with data associated recurrent neural networks. *CoRR*, abs/1703.03098, 2017.

[XGD+17] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 5987–5995, July 2017.

[YLL+18] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1168–1174, Oct 2018.

[ZVSL17] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.

Learning Safety-Aware Policy with Imitation Learning for Context-Adaptive Navigation

Bo Xiong[1], Fangshi Wang[1], Chao Yu[2], Fei Qiao[3,*], Yi Yang[3], Qi Wei[3] and Xin-Jun Liu[2]

[1]School of Software Engineering, Beijing Jiaotong University, Beijing, China
[2]Department of Mechanical Engineering, Tsinghua University, Beijing, China
[3]Department of Electronic Engineering, Tsinghua University, Beijing, China
[*]Corresponding author: qiaofei@tsinghua.edu.cn

Abstract

This paper presents an Imitation Learning (IL) based visual navigation system, which could guide the robots navigating from some start position to a goal location without any explicit map. We pay close attention to the safety issue due to partially-observability and data distribution mismatching—when the robot meets some incomplete or unfamiliar states, it probably performs an unsafe action, making it hard to work on lifelong robot navigation. In this paper, a sequence-to-sequence (Seq2seq) deep neural network is built to enhance the agent's context-awareness in partially-observable conditions and boost the model's adaptability to unseen scenarios. Additionally, we propose Uncertainty-Aware Imitation Learning (UAIL) by explicitly estimating model uncertainty and actively request experts for labeling samples according to the uncertainty with On-Policy IL. Simulations demonstrated that the combined method—Safety-Aware Imitation Learning (SAIL) in goal-driven visual navigation achieves 35.6% shorter expected moving steps and 22% fewer collisions compared with current counterparts. With the learned safer policy, SAIL had be successfully adapted to unseen environments with minimal navigation performance loss.

1 Introduction

Considering a task of navigating from a current location to find a specific goal. Classical geometry-based methods, such as Simultaneous Localization and Mapping (SLAM) [1], [2], [3], [4], [5] can be divided into two stages: one stage is building a 3D map using imagery feature matching and geometry constraints, the other stage is global or local path planning. SLAM-based approaches require carefully designed image features and are hard to work on texture-less environments. Recently, learning-based approaches have dominated robot learning including manipulation [21, 22], self-driving cars [10, 15, 16] and robot navigation [19, 20]. Compared with traditional methods, learning-based navigation could work in an end-to-end fashion without any explicit map-building. One framework for doing this is Reinforcement Learning (RL) [6, 7, 8, 9]. A reward function is usually given in RL, then agents learn a policy to maximize the cumulative reward by interacting with the environment. However, designing such an appropriate reward function in real-world scenarios is too difficult for humans. Moreover, the sparse reward of RL in goal-driven task could lead to poor convergence and computation efficiency.

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

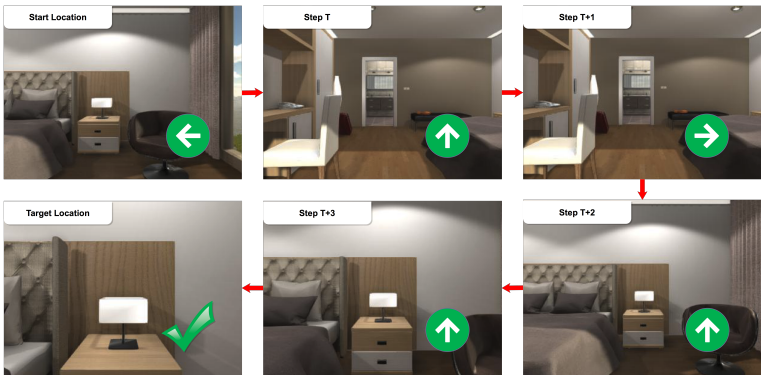


Figure 1: End-to-end target driven navigation task, the goal is to learn a mapping from observation to action which consumes minimum moving steps from a start position to the target location without any known map.

Alternatively, Imitation Learning (IL) [10, 11, 12] has been proposed to resolve reward-function-designing issues in RL. Rather than designing such a reward function, IL could learn policies directly from observing expert's demonstrations and generalize to new situations without any explicit interaction with environments. A common approach for IL is Behavior Cloning (BC) [13], where a robot observes a supervisor's policy and learns a mapping from states to actions directly with supervised learning. However, BC has a prerequisite that demonstrations must meet the i.i.d assumption of statistical learning, or will suffer from several problems—with the execution of the robot's policy, robot's state will move to a different distribution from teacher's demonstration which it was trained on, making it drift to dangerous states [14]. For instance, when a robot moving to a strange state, collisions could easily happen. Moreover, the robot's action estimation errors will compound once the robot's states drift away from the supervisor's demonstrations. Therefore, the model is hard to be adapted to context-changing environments, which prevents the application in life-long navigation. On-policy approach, such as Data Aggregation [14], can partially alleviate this issue by querying corrective samples online and iteratively aggregate new data for training. However, DAgger requires a huge number of queries to update its policy, which could be tedious for human teachers to answer and add more unnecessary computations. Recent works have paid more attention to safety issues in robot learning tasks, such as safe RL [43, 44, 45], but seldom consider the IL's safety in specific applications. In addition, the model's adaptability toward unseen environments plays a vital role in lifelong robot navigation, which should be explicitly modeled in the deep neural network of IL.

This paper presents Safety-Aware Imitation Learning (SAIL) framework by addressing both partially observability and data distribution mismatching in IL. Firstly, to enhance the context-awareness in partially observable environments and facilitate the adaptability to unseen scenarios and goals, we build a sequence-to-sequence deep neural network with Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). In this network, both spatial relevance and temporal relevance are taken into consideration, which could significantly enhance the agent's context-awareness and improve the model's generation performance toward unseen scenarios and goals. Secondly, UAIL is proposed using Bayesian approximation with MC-Dropout [42]. For those potentially uncertain or unsafe actions that occurred in some unseen or unfamiliar scenarios, rather than to perform it anyway, UAIL request for expert's advising (similar to active learning) for whether to act or label it. We predict the model uncertainty with MC-Dropout, a Bayesian approximation for uncertainty estimation in deep learning. This uncertainty has been used to improve the safety and efficiency of On-Policy Imitation Learning such as DAgger. Extensive experiments in the simulator have been conducted to compare the combined SAIL method with the vanilla IL approach, SAIL shows better navigation performance (35.6% fewer expected moving steps) and safer policy (22% lower collision rates). Additionally, with the learned safer policy, SAIL had shown successfully adapted to unseen scenarios and goals with minimal navigation performance loss. The main contributions of our work are listed as followings:

(1) We built a Seq2seq deep neural network to enhance the agent's context-awareness in partially-observable scenarios and facilitate the model's adaptability to unseen scenarios and goals.

(2) We presented Uncertainty-Aware Imitation Learning (UAIL) approach by estimating model uncertainty with MC-Dropout and combining it with On-policy IL method, which significantly improves the safety of IL.

(3) We proposed SAIL by combining UAIL and the Seq2seq network and done extensive experiments and evaluations, including navigation performance, safety, and the adaptability to unseen environments and goals.

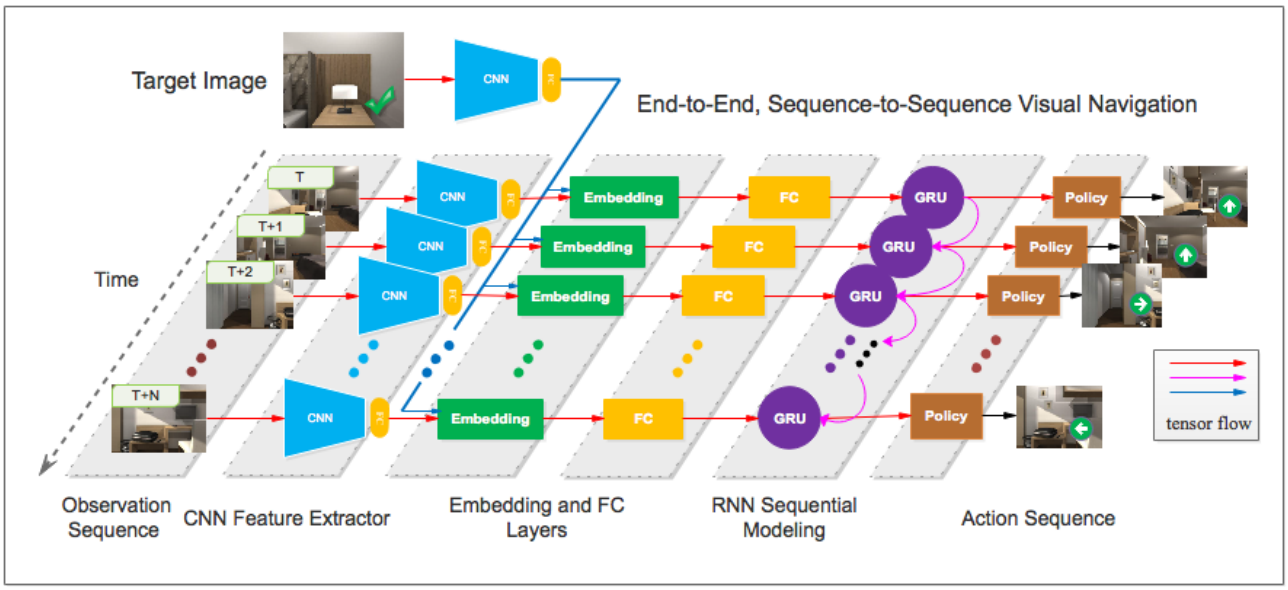


Figure 2: The proposed end-to-end, sequence-to-sequence neural network architecture for context-adaptive visual navigation. 1) we model the spatial relevance between observation and goal with Siamese CNN Network. in each time step, both of current observation and goal image are feed into the CNN network, where ResNet-50 are used to extract the visual features of images. 2) we model the temporal relevance between current observations and past observation it has seen before to address partially observability. GRU is used to maintain the hidden representations of the input observations it has seen before.

In the following sections, related work is presented in Section II. Section III demonstrates the SAIL framework. Experimental results are described in Section IV before making a conclusion in Section V.

2 Related Work

This work is relevant to past literature in the domains of imitation learning, learning-based navigation and uncertainty estimation. In this section, these areas' related work is reviewed respectively.

2.1 Imitation Learning

One of the commonly used solutions for IL is Behavioral Cloning (BC) [13], where the robot passively observes expert's full demonstrations and learns a policy mapping state to action via purely supervised learning. However, BC suffers from serious safety problems, when executing its policy, the robot will drift to dangerous states. For example, when a self-driving car steers to the edge of the road, it cannot be able to recover from it [14]. [23] has pointed out it was due to the robot's distribution being different from its demonstrator's, once robot drifting away from expert's demonstrations, robot's error will compound, which is a known problem named compounding error (or covariate shift). DART algorithm [24], where noise is injected into the expert's trajectory, can make the gap of distribution more nearly, but DART cannot work on the scenario where the gap is serious. Inverse Reinforcement Learning (IRL) [25] could restore the reward function from the expert's behavior trajectory, which enables RL in turn. [26], [27] are two different branches of IRL. Generative Adversarial Imitation Learning (GAIL) [28] build a generator and a discriminator to find a strategy that matches the distribution of state-action pairs of experts and does not require any assumptions about the environment. The common problem of IRL and GAIL is their poor scalability in real-world settings and expensive computation. On-policy approaches have been proposed to address the compounding error problem in off-policy IL. DAgger [14], [15] (Data Aggregation) is one of the classical on-policy solutions. By continuously querying the experts for new corrections during execution, it can make the robot's execution trajectory distribution closer to the supervisors'. which has been demonstrated to reduce compounding error and learn robust policies. However, DAgger suffers from several limitations: 1) it is difficult for human experts to provide enough labeling; 2) visiting highly sub-optimal states is potentially dangerous for a robot in real settings [29]; 3) it is computationally expensive to iteratively update the policy.

To alleviate the computation burden, [29] tries to train a classifier to predict whether or not is safe during the robot's execution. However, this approach adds challenges to computational efficiency and human experts.

2.2 Learning-based Navigation

Learning-based techniques, especially the deep learning, approach the visual navigation problem in an end-to-end fashion. There are mainly two types of learning based visual navigation methods: 1) RL based methods [6, 7, 8, 9]: RL based methods are divided into two stages: an exploration stages, where the environment's map information is implicitly gathered; and an exploitation step, where the map information is used to navigate efficiently [30], The paper [6] explored deep RL for target-driven navigation—navigating from a current location to a target position. The main objective of goal-driven navigation is to find the minimal sequence of actions from its current location to a target. However, this method requires 100 million frames to converge, which is too difficult to train; [30] applied an LSTM extension to DRL for 3D maze navigation. 2) IL-based methods: [32] proposed an imitation learning method for autonomous control of an aerial vehicle. [33] explored 3D navigation tasks such as 2D grid navigation, target-reaching and line-following with deep IL. [32] applied the IL to autonomous navigation in complex natural terrain. [34] proposed zero-shot IL for visual navigation. All of the IL-based methods suffer from either unsafe problem due to data distribution mismatch or poor computation efficiency.

2.3 Uncertainty Estimation in Deep Learning

There are two types of uncertainty in deep learning: 1) Aleatoric uncertainty, or statistical uncertainty, is resulted from the intrinsic data distribution themselves [46,47]. There are some approaches [48] had been used to model aleatoric uncertainty, but it cannot be alleviated by simply adding more samples or prior knowledge to the deep learning system. 2) Epistemic uncertainty, as known as systematic uncertainty, come from the model parameters themselves due to limited it theoretically could be eliminated by giving additional training samples or offering more prior knowledge. [50] provide a deep analysis of estimating uncertainty in deep neural networks. One of the most popularly used frameworks for estimating epistemic uncertainty in deep learning is Bayesian approximation. Such as MC-dropout [46, 49] and Bayesian ensemble [51]. In this work, we applied MC- Dropout to estimate the model uncertainty due to its plug-and- play advantages in existing deep network architectures. The only thing we need to do is to pass the same input to the deep network multiple times with random dropout rate and compute the entropy or variance of the outputs, which could be used to represent the uncertainty.

3 SAIL METHOD

In this section, the SAIL framework for context-adaptive navigation is described in detail. Firstly, we introduce context-aware deep neural network design for goal-driven navigation, then the safety aware imitation learning using uncertainty estimation with MC-Dropout is presented.

3.1 Context-Aware Neural Network Design

For network design, as shown in Fig.2, we consider the spatial relevance between the observations and goal image. In each time step, both of the current observation and goal image are fed into the network. The intuition of doing this is inspired by [6], [40], which can be used to improve the model's generalization performance on unseen goals. When applying the IL approach to unseen goals, the network needs no re-training for each goal. Furthermore, we also take into account the temporal relevance of state-action pairs before the current observation. This technique works especially on scenarios where the robot can hardly observe all of the details of the current state (or Partially-Observable Markovain Decision-Making). Recurrent Neural Network (RNN) is used to address this problem. The goal of the network is to estimate the current action (such as moving forward or turning right) from all of the history states and goal together. To be specific, this network can be divided into three parts.

CNN Feature Extractor: To extract the image features of observations and targets respectively. we use two separate weights-shared CNN streams—ResNet-50 [35]. which are used to transform the two images into the same embedding space. ResNet-50 is pre-trained on ImageNet [36] and finetuned on our dataset before the training stage. The outputs of the ResNet-50 are projected into a 512-dimension space.

Embedding and FC Layer: this layer is used to fuse the observation feature and goal image feature to a 1024- dimension joint representation and then projected it to a new 512-dimension vector on fully-connected fusion layers. We use two separate fully-connected layers to learn the joint feature representation of observation and goal image.

RNN Context Modeling: To learn the temporal relevance of observation sequences. RNN Layers is added right after the embedding and FC layer. Gradient Vanishing is a key problem of RNN with long-term dependencies. Although LSTMs [38] are more prevalent in addressing this problem in past literature, we make use of GRUs [37] that have smaller number of parameters, simpler to use and are generally faster to train than LSTMs.

A sequence-to-sequence IL model is built as Fig 2. At the time step $t = 0$, an initial goal g and a start state s_0 are fed into the network, then in each time step $t > 0$, the agent takes an action at with its current policy π_θ.

$$a_t = \pi_\theta(s_t, g) \tag{1}$$

Then the environment changes its state to a new state s_{t+1} according to the transition dynamic env.

$$s_{t+1} = env(s_t, a_t, g) \tag{2}$$

The current hidden vector h_t is a function f_θ on current observation st, current goal g and the previous hidden vector h_{t-1}, θ is the parameter of function f_θ.

$$h_t = \begin{cases} s_0 & t = 0 \\ f_\theta(h_{t_1}, s_t) & t >= 1 \end{cases} \tag{3}$$

We can predict an optimal action $a_t^\wedge$ with $P_\theta(a_t|h_t)$.

$$a_t^\wedge = arg \max_a P_\theta(a_t|h_t) \tag{4}$$

where

$$P_\theta(a_t|h_t) = softmax(h_t) \tag{5}$$

Given the training set $D = (s^i, a^i)$ The goal is to maximize the log-likelihood of the output action sequences.

$$\theta^* = arg \max_\theta log \sum_{(s^i,a^i)\in D P_\theta(a^i|s^i)} \tag{6}$$

where

$$\log P_\theta(a|s) = \sum_t \log P_\theta(a^t|s^t) \tag{7}$$

We minimize the cross-entropy loss between predicted action and corrective action made by experts with Adam optimization algorithm.

$$L(\theta) = \sum_{\tau in D} \sum_a [\pi_E(a|s, g) \log(\pi_\theta(a|s, g))] \tag{8}$$

3.2 Safety-Aware Imitation Learning

Considering the safety issues of off-policy IL due to the data distribution mismatching. UAIL was proposed by considering model uncertainty and combining it with vanilla on-policy training methods—Data Aggregation (DAgger) [14], [15]. UAIL iteratively query the expert but not frequently as DAgger for new correctives samples to train IL model. The uncertainty information in the current model is used to guide UAIL whether or not to ask the expert for new sample labeling. Therefore, this combined on-policy IL method is query-efficient, only when the uncertainty value is greater than a certain threshold, then we ask the expert for labeling and vice versa. when the accumulated new labeled data is enough for training a new policy (the number of new labeled data is larger than another threshold), we aggregate the new labeled data with the old one and retraining our network using the new aggregated dataset. A basic idea to estimate the model uncertainty in neural networks is using the entropy of softmax output in the categorical neural network. This method is very naïve because it always happened that the softmax entropy is large but the real uncertainty is small and vice versa.

Uncertainty Estimation with MC-Dropout: To bridge the gap between uncertainty estimation and deep neural network, we use MC-Dropout, a Bayesian approximation of uncertainty estimation in deep learning to represent the confidence of action to be executed. Basically, it has been demonstrated that using Dropout

at inference time in the deep neural network is equivalent to doing Bayesian approximation in Bayesian deep learning. The key idea here is letting dropout doing the same thing in both training and testing time. At test time, we will repeat *beta* times in passing the same input to the network with a random dropout value. MC-Dropout provides an efficient way to estimate uncertainty with minimal changes in most existing deep networks. It provides a plug-and-play module to deep learning for uncertainty estimation.

Safety-Aware Imitation Learning: To improve the training efficiency of the on-policy training approach, we intend to reduce the number of queries and aggregation times for data aggregation. We propose an uncertainty-aware imitation learning method (similar to active learning). It initializes model's weights on initial dataset with supervised learning, then executes the current policy until the learner's confidence falls below a solid threshold, at which point it queries the expert for corrective action. Uncertainty based approach may decide to stop asking for queries once the confidence exceeds the threshold in all states. which make use of the complementary advantages of humans and robots. Algorithm 1 summarizes the training procedure of Safe IL with uncertainty estimation.

Algorithm 1 SAIL: Safety Aware Imitation Learning.

Input: Initial demonstrations, D_0; uncertainty threshold, $\epsilon_{uncertainty}$; data aggregation threshold $\epsilon_{aggregation}$; aggregation episodes, M; batch size,N;

Output: policy parameter, θ;

```
θ_0 = SupervisedImitationLearning(D_0);
for I = 0; M do
    for t = 0; N do
        a_{t+1,θ} = PolicyExecute(π_{θ_t}, s_{i,t}, g)
        ε_uncertainty = MCDropout(a_t, s_{i,t}, g)
        if a_t < ε_uncertainty then
            a_{t+1,E} = QueryExpert(s_{i,t}, g)
            D_t, T = AggregationPool(a_{t+1,E}, s_{i,t}, g)
        else
            if T > ε_aggregation then
                D_{t+1} = DataAggregation(D_0, D_t)
                break
            else
                return (4)
            end if
        end if
    end forSupervisedImitationLearning(D_{t+1})
end for
returnθ
```

4 Experiments

In this section, we evaluate the SAIL approach in the simulation environment—AI2-THOR1 [39]. We start out by introducing the dataset and experiment setup, then present extensive experimental results on several metrics.

4.1 Dataset

AI2-THOR framework is used to collect the training data. AI2-THOR is an excellent photo-realistic interactive 3D simulator for AI agents, it provides a 3D environment that looks similar to the real-world scenes. There are totally 120 scenes in the AI2-THOR environment covering 4 different environment types, including kitchens, living rooms, bedrooms, and bathrooms, and each room type consists of 30 specific scenes. In AI2-THOR, the agent's positions are sampled from the discrete grids with 4 action spaces (move forward, move back, turn right and turn left). Each image consists of the agent first-person view in the rooms.

4.2 Experiment Setup

In this experiment, the authors use 4 different scenes to train our models (bathrooms, bedrooms, kitchens, and living rooms). For expert's policy generation, we use the A* search algorithm to find the shortest path from the start location to the target location while avoiding collisions simultaneously. In order to narrow the situation

gap between different room types, we trained our model on each scene separately. In each scene, the authors set 10 remarkable goal locations, such as the laptops, chairs and table lamps. For each goal, this work runs 50 stochastic starting points to evaluate the performance. To validate the proposed approach, the authors compare and evaluate the following 4 baselines and 2 proposed methods.

(1) BC (off-policy): an off-policy IL algorithm that learns a mapping from states to actions directly by supervised learning without any demonstrations sampling.

(2) DART (off-policy): an improved BC method that randomly injects noise into the expert's trajectory.

(3) DAgger (on-policy): an on-policy training approach by iteratively querying experts for the new correct sample and aggregating the new dataset for training.

(4) SaferDAgger (on-policy): an extension of vanilla DAgger method that minimizes the number of queries to a reference policy both during training and testing stage.

4.3 Evaluations and Results

The ultimate goal of goal-driven navigation is to find a given target with minimum steps while avoiding collisions simultaneously. Although these two facts can be affected by each other, having fewer moving steps is not equal to having fewer collisions in real scenarios. Therefore, we evaluate SAIL's performance from several different metrics.

Expected Trajectory Steps. We evaluate the navigation performance by expected trajectory steps, the expected trajectory steps are defined as the total number of steps taken to navigate from an initial start position to a given target location. The goal of this task is to minimize the expected trajectory steps, the closer the expected navigation steps to the shortest path is, the better the navigation's performance is. For four different scenes, the authors compare the training results of 4 different scenes with shortest path steps, as shown in Table 1. It can be seen that SAIL with MC-Dropout achieve shorter expected moving steps than vanilla off-policy and on-policy IL approaches.

Navigation Safety Evaluation. In order to evaluate the robot's safety awareness in goal-directed navigation, the expected collision rate is defined as the mean percentages of robot's collisions with obstacles when navigating from different start locations to different target positions. In this work, the agent is allowed to collide with the environment, when the agent collides with the environment, we recognize it as one navigation step. As shown in Table 2. SAIL with MC-Dropout can significantly reduce the collision rate of goal-driven navigation.

Navigation Success Rate in Unseen Scenes. Another metric to evaluate navigation performance is the navigation success rate in unseen scenes. Since there are 120 different scenes in each types of environments, we set 100 scenes as training scenes and 20 scenes as testing scenes. Experiment result (see Table 3) show that the performance of SAIL in unseen scenes was farther markedly improved than the baseline IL model. This can be explained by the safer policy learned by SAIL and improved perception ability due to context-aware neural network design.

Table 1: Expected moving steps and collision rate for goal-driven navigation.

Method	Bathroom	Kitchen	Bedroom	Living room
Shortest Path	7.11	10.87	15.37	16.02
BC [13]	15.58	37.19	55.95	66.34
DART [24]	15.51	36.62	53.38	63.21
DAgger [14]	14.31	35.52	51.39	55.21
SaferDagger [29]	13.34	25.01	46.46	48.31
SAIL(Entropy)	13.34	25.01	46.46	48.31
SAIL (MC-Dropout)	12.21	17.35	33.55	37.32

5 Conclusion

This work proposes the SAIL approach for goal-driven context-adaptive visual navigation. This approach addresses the robot's safety issues due to data distribution mismatch and poor performance on partially-observable visual navigation. To alleviate the safety issue, we model the policy uncertainty in deep learning with MC-Dropout and combine it with the on-policy IL approach. To enhance the agent's perception capability and context awareness in partially-observable environments, we build a sequence-to-sequence deep neural network

Table 2: Expected collision rate for goal-driven navigation.

Method	Bathroom	Kitchen	Bedroom	Living room
BC [13]	0.30	0.43	0.49	0.52
DART [24]	0.28	0.41	0.42	0.45
DAgger [14]	0.29	0.42	0.45	0.47
SaferDAgger [29]	0.26	0.38	0.41	0.44
SAIL(Entropy)	0.26	0.32	0.39	0.41
SAIL (MC-Dropout)	0.24	0.31	0.35	0.37

Table 3: Navigation success rate in unseen scenes,where the maximum penalty factor is 5 times the steps of shortest path.

Method	Bathroom	Kitchen	Bedroom	Living room
BC [13]	0.89	0.83	0.71	0.64
DART [24]	0.91	0.87	0.74	0.67
DAgger [14]	0.88	0.86	0.74	0.68
SaferDagger [29]	0.96	0.91	0.81	0.69
SAIL(Entropy)	0.95	0.94	0.89	0.75
SAIL (MC-Dropout)	0.96	0.98	0.92	0.81

with both spatial and temporal relevance taken into consideration. Experiments on simulator demonstrated that the proposed SAIL method had better navigation performance on several evaluation metrics, compared with the state-of-the-art IL methods. With the safer policy, SAIL had shown successfully be adapted to unseen scenarios and goals with fewer collisions and minimal navigation performance loss.

Acknowledgement

We are thankful to Ai Shi who moderated this paper and, in that line improved the manuscript significantly.

References

[1] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.

[2] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007 (6): 1052-1067.

[3] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]//European Conference on Computer Vision. Springer, Cham, 2014: 834-849.

[4] Endres F, Hess J, Sturm J, et al. 3-D mapping with an RGB-D camera[J]. IEEE Transactions on Robotics, 2014, 30(1): 177-187.

[5] Yu, Chao, Liu, Zuxin, Liu, Xinjun, et al. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments[J]. 2018.

[6] Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning[C]//Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017: 3357-3364.

[7] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning, 2016

[8] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. Deep reinforcement learning with successor features for navigation across similar environments. arXiv preprint arXiv:1612.05533, 2016.

[9] J. Zhang, L. Tai, Y. Xiong, M. Liu, J. Boedecker, and W. Burgard. Vr goggles for robots: Real-to-sim domain adaptation for visual control. arXiv preprint arXiv:1802.00265, 2018.

[10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. arXiv preprint:1604.07316, 2016.

[11] Giusti A, Guzzi J, Ciresan D C, et al. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots[J]. IEEE Robotics and Automation Letters, 2016, 1(2): 661-667.

[12] N. D. Ratliff, J. A. Bagnell, and S. S. Srinivasa. Imitation learning for locomotion and manipulation. In International Conference on Humanoid Robots, 2007.

[13] Michael Bain and Claude Sommut. A framework for behavioural claning. Machine Intelligence 15, 15:103, 1999.

[14] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. arXiv preprint arXiv:1011.0686, 2010.

[15] Codevilla F, Müller M, Dosovitskiy A, et al. End-to-end driving via conditional imitation learning[J]. arXiv preprint arXiv:1710.02410, 2017.

[16] Pan Y, Cheng C A, Saigol K, et al. Agile Off-Road Autonomous Driving Using End-to-End Deep Imitation Learning[J]. arXiv preprint arXiv:1709.07174, 2017.

[17] Giusti A, Guzzi J, Ciresan D C, et al. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots[J]. IEEE Robotics and Automation Letters, 2016, 1(2): 661-667.

[18] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert. Learning monocular reactive UAV control in cluttered natural environments. In ICRA, 2013.

[19] Pathak D, Mahmoudieh P, Luo G, et al. Zero-shot visual imitation[C]//International Conference on Learning Representations. 2018.

[20] Tai L, Zhang J, Liu M, et al. Socially-compliant navigation through raw depth inputs with generative adversarial imitation learning[J]. arXiv preprint arXiv:1710.02543, 2017.

[21] N. D. Ratliff, J. A. Bagnell, and S. S. Srinivasa. Imitation learning for locomotion and manipulation. In International Conference on Humanoid Robots, 2007.

[22] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth. Model-based imitation learning by probabilistic trajectory matching. In ICRA, 2013.

[23] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In International Conference on Artificial Intelligence and Statistics, pages 661–668, 2010.

[24] Laskey M, Lee J, Fox R, et al. Dart: Noise injection for robust imitation learning[J]. arXiv preprint arXiv:1703.09327, 2017.

[25] Abbeel P, Ng A Y. Inverse reinforcement learning[M] //Encyclopedia of machine learning. Springer, Boston, MA, 2011: 554-558.

[26] Ziebart B D, Maas A L, Bagnell J A, et al. Maximum Entropy Inverse Reinforcement Learning[C]//AAAI. 2008, 8: 1433-1438.

[27] Ramachandran D, Amir E. Bayesian inverse reinforcement learning[J]. Urbana, 2007, 51(61801): 1-4.

[28] Ho J, Ermon S. Generative adversarial imitation learning[C]//Advances in Neural Information Processing Systems. 2016: 4565-4573.

[29] J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. arXiv preprint arXiv:1605.06450, 2016.

[30] Dhiman V, Banerjee S, Griffin B, et al. A Critical Investigation of Deep Reinforcement Learning for Navigation[J]. arXiv preprint arXiv:1802.02274, 2018.

[31] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. 2017.

[32] Sammut C, Hurst S, Kedzier D, Michie D et al (1992) Learning to fly. In: Proceedings of the ninth international workshop on machine learning, pp 385–393

[33] Hussein A, Elyan E, Gaber M M, et al. Deep imitation learning for 3D navigation tasks[J]. Neural computing and applications, 2018, 29(7): 389-404.

[34] Silver D, Bagnell J A, Stentz A. Applied imitation learning for autonomous navigation in complex natural terrain[C]//Field and Service Robotics. Springer, Berlin, Heidelberg, 2010: 249-259.

[35] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//AAAI. 2017, 4: 12.

[36] Olga Russakovsky, Jia Deng, Hao Su, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3):211-252.

[37] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures[C]// International Conference on International Conference on Machine Learning. JMLR.org, 2015:2342-2350.

[38] Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling[C]//Fifteenth annual conference of the international speech communication association. 2014.

[39] Kolve E, Mottaghi R, Gordon D, et al. AI2-THOR: An interactive 3d environment for visual AI[J]. arXiv preprint arXiv:1712.05474, 2017.

[40] Sadeghi, F., Toshev, A., Jang, E., & Levine, S. (2018). Sim2Real Viewpoint Invariant Visual Servoing by Recurrent Control. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4691-4699).

Guidance of Mobile Robot Navigation in Urban Environment using Human-Centered Cloud Map

Jae-Yeong Lee, Sunglok Choi, Seunghwan Park, Jaeho Lim, Seungmin Choi, Seohyun Jeon, Yunseok Lee, Beomsu Seo, Wonpil Yu
Intelligent Robot Research Laboratory, ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, Republic of Korea
jylee@etri.re.kr

Abstract

Autonomous navigation in a city-scale environment brings several technical challenges that are difficult to solve by traditional approaches. In this paper, we briefly discuss the limitations of the conventional navigation methods based on robot-centered environment modeling and understanding, and present recent an ongoing developments of the DeepGuider Project. The DeepGuider Project aims to develop a navigation guidance system that enables robots to navigate in urban environment without pre-mapping of the environment. In the paper, the main concepts and overall system architecture is briefly presented.

1 Introduction

Autonomous navigation in the unconstrained environments is one of the most important functions in realizing robot services. In particular, there has been increasing demand on service robots in urban and everyday living environments such as courier robots and food delivery robots. In order to enable such robot services, a robot navigation technology capable of navigating to an arbitrary destination in the urban environment is required.

Autonomous navigation in a city-scale environment brings several technical challenges that are difficult to solve by traditional methods. First, traditional robot navigation requires a precise robot-centered map which is usually built in advance through SLAM(simultaneous localization and mapping) algorithm. However, robot-centered modeling or mapping of city-scale space including every streets and buildings is very costly and sometimes impractical. Second, even after initial mapping is successful, detecting changes (static changes like new buildings, shops, trees) and keeping the map up to date for consistency is another challenging issue. The environment is not fixed and the dynamic and static changes cause significant problems for existing map-based approaches. Third, the success of map-based navigation relies on accurate localization. However, in a complex and dynamic urban environment, accurate and reliable localization is easy to fail, leading to a navigation failure.

These problems are difficult to avoid with current approaches of robot-centric environment modeling and understanding. On the other hand, unlike robots, we humans have the ability to navigate even in unvisited cities and places. It is mainly because human is able to understand and utilize semantics of the surroundings (roads, buildings, the connection of roads, landmarks) with an optional aid of abstracted map information (paths and landmarks from electronic maps such as Google Map, Naver Map).

In this paper, we introduce main concepts of the DeepGuider Project which started recently as a national research project in Korea, and describe the technical issues and the proposed system architecture. The DeepGuider

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Project aims to develop a navigation guidance system that enables robots to navigate in indoor and outdoor urban environments without pre-mapping of the environment nor any pre-built robot-centered map. Instead of robot-centered map, the guidance system utilizes existing human-centered digital maps such as Google Map or Naver Map (hereinafter, they are called cloud map) to get abstracted navigation information of the environment. The abstract navigation information includes road topology, path to destination, and POIs[1] along the path. Street-view or road-view images provided by the cloud map services and GPS information can also be optionally utilized.

Main advantages of the DeepGuider approach is as follows. Since the proposed system uses existing human-centered navigation maps, there is no need for additional mapping and it is possible to apply a robot navigation service instantly to any places and areas. Therefore, if the proposed system is realized, nationwide navigation service is possible, and various indoor and outdoor robot services such as delivering goods and guiding people to places can be realized. The DeepGuider Project is an open source software project, and its all results are released in public via a GitHub repository (https://github.com/deepguider).

2 Related Works

There have been many studies on minimizing mapping efforts or mapless navigation to overcome the limits of traditional SLAM-based navigation. Brubaker *et al.* [1] proposed a self-localization method which utilizes visual odometry and online road maps as the inputs. It localizes by matching the shape of trajectory of the vehicle obtained from visual odometry with the ones from free online OpenStreetMap. They adopt a probabilistic approach to cope with inherent ambiguities in the map (*e.g.,* in a Manhattan world). Recently, Mirowski *et al.* [2] presented an end-to-end deep reinforcement learning approach that can be applied on a city scale. They show that it is possible to learn navigation directions by using only Google StreetView without pre-given map. It demonstrates large-scale learning from real-world imagery, but training and testing is done on the same environment. Google also recently announced concept of experimental research of global localization, which combines Visual Positioning Service (VPS), StreetView, and machine learning to accurately identify position and orientation in urban environment[4]. It uses the smartphone camera as a sensor and Google StreetView images as references to match. The problem is that the imagery from the phone at the time of localization may differ from what the scene looked like when the Street View imagery was collected. As one way, they suggest to filter out temporary parts of the scene and focus on permanent structure that doesn't change over time by machine learning automatically.

Another branch of approach is topological representation of the space and localization. Milford *et al.* [3] proposed the RatSLAM method based on the rat's navigation mechanism. RatSLAM builds a local graph map of the nodes of spaces in online and localizes based on the topological connectivity of the spaces and feature matching of each space. Badino *et al.* [5] proposed a hybrid topometric localization method that combines topological localization using spatial connectivity of the places and metric localization method by Bayesian filtering. Recently, Bruce *et al.* [6] presented a reinforcement learning method that learns navigation controls to reach destination based on a topological representation of the space with omnidirectional images as nodes of the navigation graph.

Road structure or topology provide an important clue for a semantic understanding of the environment and localization. However, there have been only a limited number of studies on this branch. Brubaker *et al.* [1], as described already, utilizes shape of road for self-localization. Kumar *et al.* [7] presented a method to classify road types on street images into intersection and non-intersection based on deep network ensembles. They reported 72.1% accuracy on Mapillary images which consists of 300,000 street images. Amini *et al.* [8] suggested a deep learning method to output vehicle control from raw sensor data and high level of route map using a variational network. Researches on extracting or recognizing road topology have been conducted mainly on aerial photos [9] and research on frontal images on the ground is very rare.

3 System Architecture

Figure 1 shows overall architecture of the proposed guidance system. The overall system flow is as follows. Once a user requests a robot service through the DeepGuider system with a destination information, the guidance system retrieves paths and map information from the cloud map service. Then the guidance system tries to recognize road structure, POIs, and other semantic information from the raw sensory inputs from the robot.

[1]Points of Interest: a specific point location that someone may find useful or interesting.

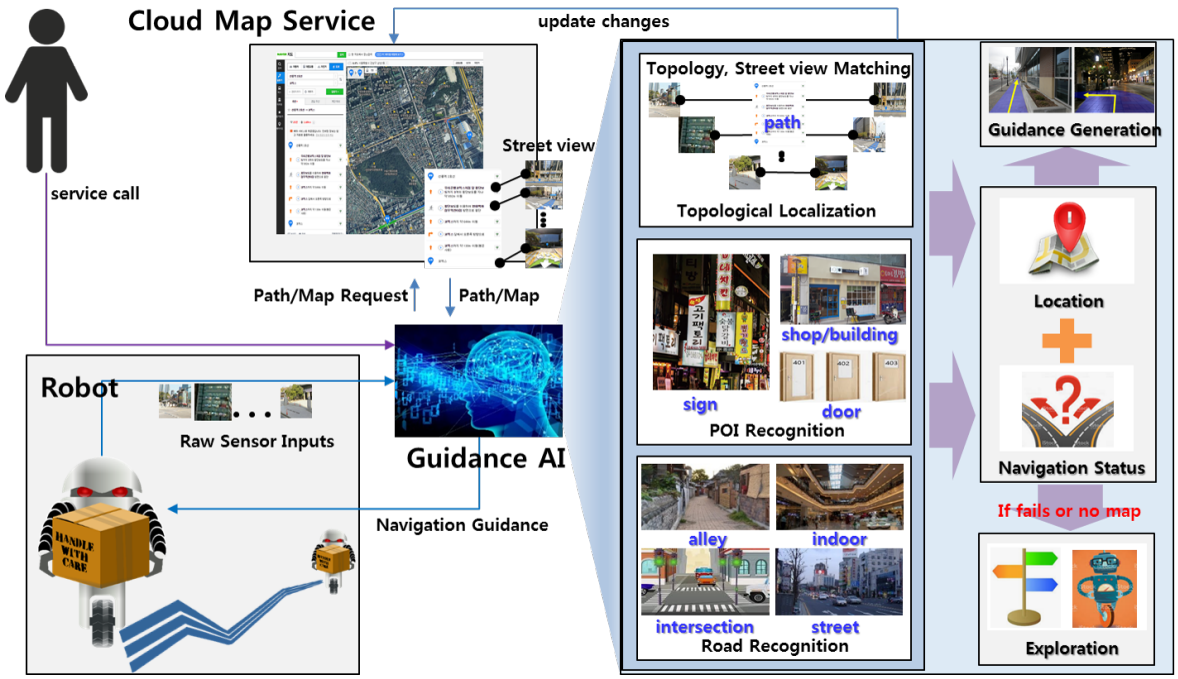


Figure 1: System architecture

The extracted information then is matched with the map information to locate robot position on the path. If the localization is successful, an online navigation guidance is generated and sent to robot. On the other hand, if the localization fails or it gets lost, the guidance system invokes an exploration module, which find ways until location is recovered.

4 Implementation

The DeepGuider system is currently in development. Therefore, only the guidance scenarios in normal and lost situation are described here.

4.1 Guidance Scenario in Normal Conditions

After a user orders a product for delivery via web, mobile or other means, the service provider checks the ordered goods, loads them on the robot, and specifies the destination of the delivery. After confirming that the delivery destination has been specified, the guidance system accesses the cloud map service and retrieves a routing path from the current position of the robot to the destination. Since the routing path obtained from the cloud map service is composed of a vehicle-centric or a pedestrian-centric path, it is difficult to directly use it for the robot navigation. The guidance system converts the routing path as a sequence of predefined robot guidance commands. The robot guidance commands consist of nodes and actions. The nodes are the important way points in the map that the robot have to pass through and the actions are the semantic motion commands to direct the robot to the next node. After that, the start command is transmitted to the robot. And during the navigation, a guidance command in every step is selected according to the position of the robot and is sent to the robot.

The robot captures the front, rear and side images and other sensor data such as GPS and odometer while navigating and send them to the guidance system. The robot also automatically avoids collisions by recognizing local obstacles. The guidance system localizes the robot on the map by comparing the image and sensor data transmitted by the robot with the map information such as street view images and POIs(Point of Interests) extracted from the cloud map service. The POIs here includes the store names and logos on the path.

Based on the estimated location of the robot, the guidance system selects and provides a guide command to transmit to the robot. If the robot's final destination is located indoor, the system guides the robot to find and access the building entrance, navigate the doorway, and reach the final destination such as a specified room or

shop. If an indoor map is provided, the map information is used. If not, the destination location is estimated and searched through POI recognition and active exploration. In this case, the guidance system generates a exploring guidance command which is described in Subsection 4.2. When the destination is reached, the delivery is finished and the robot calls the user to pick up the goods.

4.2 Fail Recovery Scenario

When the robot passes a congested area or a point where it is difficult to extract feature points, the guidance system is easy to lost. For example, the robot can enter wrong alley in a complex city environments. In such cases, the guidance system recognizes that failures when a measure of reliability on the currently recognized location falls below a predefined threshold. The guidance system then propagates the context information to the internal active exploration module, and the active exploration module first attempts to return to the last successfully localized node, using the internal visual memory stored in the robot.

To return to the last successfully localized node, a guidance command utilizing visual memory is generated from the active exploration module and transferred to the robot. After the robot successfully returns to the recent node, the guidance system changes back its status to normal and resumes the normal guidance that was originally performed. If it is difficult to return to the previous node based on visual memory due to sensor uncertainty or changes in surrounding conditions, the active exploration module executes a full exploration mode. In this case, the robot tries to search in new surrounding environment until it recognizes a particular POI or node.

Even in the above two situations, the robot continuously transmits information to help the guidance system to locate the robot. And if the reliability of the current robot's position returns back to be high, the guidance system determines that the failure situation has been overcome, terminates the exploration mode, and proceeds with the normal guidance.

5 Conclusion

In this paper, we presented a new navigation framework to enable robots to navigate in urban environment without pre-mapping of the environment. The key idea is to make the robots to understand and utilize the human-centered maps or models of the environments. As the project has just started, only the concept and overall system architecture is presented in the paper. Its implementation and validation in real environment will be presented in future work.

Acknowledgement

This work was supported by the ICT R&D program of MSIT/IITP. [2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments].

References

[1] Brubaker, Marcus A., Andreas Geiger, and Raquel Urtasun. "Lost! leveraging the crowd for probabilistic visual self-localization." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.

[2] Mirowski, P., Grimes, M., Malinowski, M., Hermann, K. M., Anderson, K., Teplyashin, D., & Hadsell, R. (2018). Learning to navigate in cities without a map. In Advances in Neural Information Processing Systems (pp. 2419-2430).

[3] Milford, M., & Wyeth, G. (2010). Persistent navigation and mapping using a biologically inspired SLAM system. The International Journal of Robotics Research, 29(9), 1131-1153.

[4] https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html

[5] Badino, Hernán, Daniel Huber, and Takeo Kanade. "Real-time topometric localization." 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012.

[6] Bruce, J., Sünderhauf, N., Mirowski, P., Hadsell, R., & Milford, M. (2018). Learning deployable navigation policies at kilometer scale from a single traversal. arXiv preprint arXiv:1807.05211.

[7] Kumar, Abhijeet, et al. "Towards View-Invariant Intersection Recognition from Videos using Deep Network Ensembles." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

[8] Amini, A., Rosman, G., Karaman, S., & Rus, D. (2018). Variational End-to-End Navigation and Localization. arXiv preprint arXiv:1811.10119.

[9] Ventura, C., Pont-Tuset, J., Caelles, S., Maninis, K. K., & Van Gool, L. (2018). Iterative deep learning for road topology extraction. arXiv preprint arXiv:1808.09814.

Semantic Information-based Reliable Autonomous Navigation in Wide Space

Taeyoung Uhm, Ji-Hyun Park, Gi-Deok Bae, Jung-Woo Lee, Young-Ho Choi,
Korea Institute of Robot and Technology Convergence Pohang, Republic of Korea
{uty,jipark, bgd9047, ricow, rockboy}@kiro.re.kr
Sang-Yong Han
Kookmin University, Seoul, South Korea
syhan@kookmin.ac.kr

Abstract

Recently, much attention has been paid to intelligent robots. Especially, autonomous navigation of robots is the most important technology and is being developed by many researchers. The autonomous navigation technology is based on the SLAM to find the position from the sensor data that is mounted robots. However, most methods find locations based on high-performance sensors in the predefined environments. This is difficult to apply to complex environments such as wide area due to the difference in locomotion and sensor performance. Therefore, efforts should be made to improve the application of limited navigation technology. In this paper, we propose a method to drive in a wide space for robots with various locomotion and sensor by semantic information based autonomous navigation method. By using semantic information, the robot recognizes the surroundings using available sensor data and performs autonomous travel. For this purpose, a semantic map for a unit space (e.g. a room, a hallway, road etc.) is generated and traveled by receiving information suitable for a robot locomotion and sensor configuration from the local server. The proposed method utilizes the semantic map to drive in the same way as a person in a large space, and can use intelligent robot driving using the property information of the object. Therefore, it is expected that industrialization of robot autonomous navigation will be promoted.

1 Introduction

Research on the autonomous mobile robot has been done steadily. Recently, robots employ various locomotion and sensors [Khazanov14]. These researches assign robots by task to perform a specific task [Amigoni05], or use semantic information to service in a limited space indoors [Lim10]. In addition, there are studies that use task management, environmental awareness, trajectory planning, decision making and terrain classification using semantic maps and ontology for robot mapping [Liu12], [Li12]. However, these robots mainly carry out autonomous study on predefined areas in a way limited to locomotion and sensor system. In this paper, we

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

propose a novel autonomous navigation method that can travel a wide area through a map suitable for semantic DB-based robot motion and sensor systems developed for intelligent robots. The proposed method simulates

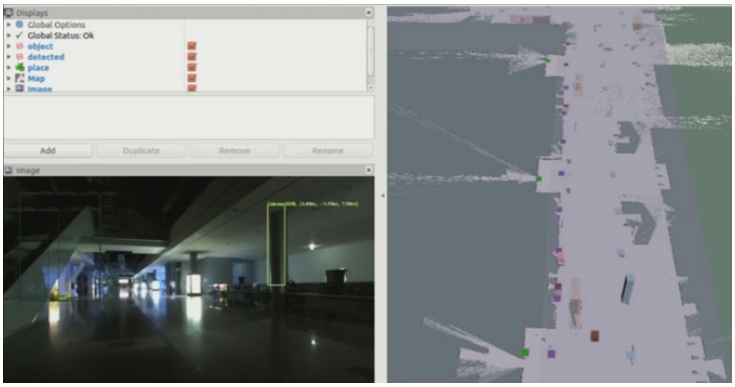


Figure 1: Semantic DB based Reliable Autonomous Navigation

the navigation method used by human beings and drives similarly. Robots travel and recognize objects around them as humans do. To do this, the robot registers and recognizes the objects necessary for navigation, as shown in 1. After that, if the recognized objects are obstacles, they will be avoided or waited after depending on the properties of the motion defined by the semantic information. If the recognized object is an object that affects driving (e.g. a rough road), the driving speed is adjusted according to the drivability defined in the object's properties.

Meanwhile, a wide-area spatial map based on 3D LiDAR sensor and vision sensor is constructed to drive wide-area space suitable for various robot motions and sensors. Robots can use this map to recognize the exact location from the multi-sensor data or even a single sensor. Therefore, the proposed driving method is useful in the wide area where many people walk because they drive according to semantic information. This is expected to bring dramatic developments to the autonomous driving of robots.

2 Navigation Method using Semantic Information

The semantic information used for autonomous driving of robots is based on objects. The robot recognizes the object while driving and uses semantic information of the object to secure driving ability similar to humans. This can be divided into two abilities: First, the motion property is used as semantic information of the object. Using this, an object, such as a person or a chair occupied by a person, is recognized as a movable object and waits for 5 seconds when the robot meets a moving obstacle while driving. After that, it is avoided in the same manner as a fixed obstacle, as shown in 2.

Figure 2: Semantic DB based Reliable Autonomous Navigation

Second, the semantic information used is the drivability property of the object corresponding to the driving road. If you are driving on a road where there is a rough property, change the speed to match the degree of

drive. The degree of roughness is divided into 5 levels and it is possible to secure the driving stability of the robot. 3 shows the difference in running speed on gravel and asphalt roads.

Figure 3: Semantic DB based Reliable Autonomous Navigation

3 Semantic DB based Map Building

The semantic DB is created by modeling object information necessary for driving from the viewpoint of the robot. Based on this, a map that is suitable for the locomotion and sensor system of the robot is build. This is a key factor for driving the robot in a wide space. 4 and 5show the entire flow chart for creating the map and the robot platform used for building the map. The platform was used to build a map based on the point cloud of the 3D LiDAR sensor and the ORB features of the vision sensor [Mur17]. 6 shows wide-area spatial maps for each sensor. It is possible to generate a semantic map including semantic information in a map suitable for the robot sensor.

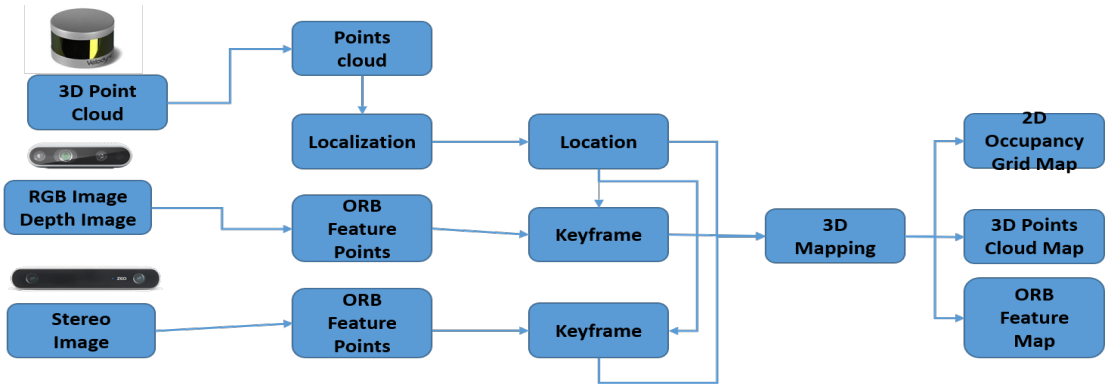


Figure 4: Semantic DB based Reliable Autonomous Navigation

Based on this semantic map, the robot performs autonomous Navigation. The robot sets up the driving strategy according to the properties of the recognized object (e.g. movable, rough etc.). Using this method, the autonomous mobile robot can flexibly move in the wide area.

4 Experimental Results

Using the proposed method, autonomous driving was carried out in a wide area of about 6ooo m2, as shown in 7.

This was tested during the exhibition at the convention center. First, the robot received information about the semantic map and mission (patrol) from the local server and started driving. Next, the robot traveled to pass between the people standing in line for entry and waited a while when it could no longer drive. The robot then continued to drive through the gaps caused by people's movement. As shown in the results, it was possible to drive smoothly in an environment with dense crowds and to travel between crowds using semantic information.

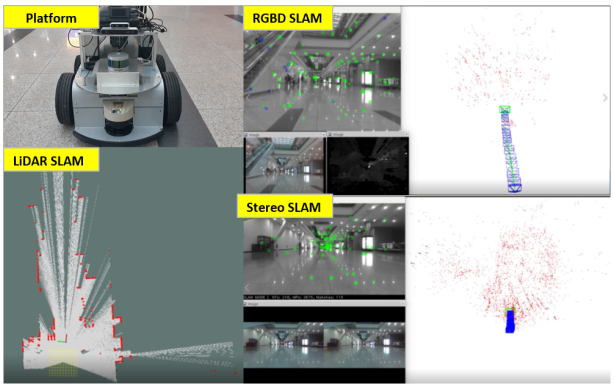


Figure 5: Semantic DB based Reliable Autonomous Navigation

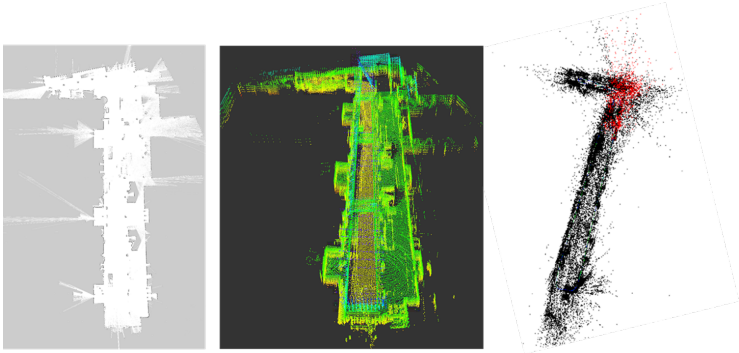


Figure 6: Semantic DB based Reliable Autonomous Navigation

Figure 7: Autonomous Navigation Results in Wide-Area

5 Conclusion and Future Work

In this paper, we propose autonomous navigation method in wide area by generating semantic map suitable for locomotion and sensor system of various robots based on the driving method performed by humans. For this purpose, driving using multi-sensor based semantic navigation map and defined semantic information was performed. In the future, we will apply semantic information-based autonomous driving methods that can be used for more types of locomotion and sensors.

Acknowledgment

This work was supported by the Korean Evaluation Institute of Industrial Technology and conducted by the Ministry of Industry and Commerce in 2017 (Industrial Core Technology Development Project, Project Number 10080489) and 2018 (Industrial Core Technology Development Project, Project Number 20000683).

References

[Khazanov14] Khazanov, Mark and Jocque, Julian and Rieffel, John, Evolution of locomotion on a physical tensegrity robot, *Artificial Life Conference Proceedings 14* – pp. 232–239, 2014.

[Amigoni05] Amigoni, Francesco and Neri, Mario Arrigoni, An application of ontology technologies to robotic agentst, *IEEE/WIC/ACM International Conference on Intelligent Agent Technology* – pp. 751–754, 2005.

[Lim10] Lim, Gi Hyun and Suh, Il Hong and Suh, Hyowon, Ontology-based unified robot knowledge for service robots in indoor environments, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* – Vol. 41, no. 3,pp. 7492-–509, 2010.

[Liu12] Liu, Ziyuan and Chen, Dong and von Wichert, Georg, Online semantic exploration of indoor maps, *2012 IEEE International Conference on Robotics and Automation* – pp. 4361–4366, 2012.

[Li12] Li, Gang and Zhu, Chun and Du, Jianhao and Cheng, Qi and Sheng, Weihua and Chen, Heping, Robot semantic mapping through wearable sensor-based human activity recognition, *2012 IEEE International Conference on Robotics and Automation* – pp. 5228–5233, 2012.

[Mur17] Mur-Artal, Raul and Tardós, Juan Dg, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, *IEEE Transactions on Robotics* – Vol 33 , no 5, pp. 1255–1262, 2017.

Towards Explanations of Plan Execution for Human-Robot Teaming

Jiyoun Moon[1] jiyounmoon@snu.ac.kr
Daniele Magazzeni[2] daniele.magazzeni@kcl.ac.uk
Michael Cashmore[3] michael.cashmore@strath.ac.uk
Dorian Buksz[2] dorian.buksz@kcl.ac.uk
Beom-Hee Lee[1] bhlee@snu.ac.kr
Yong-Seon Moon[4] moon@sunchon.ac.kr
Sang-Hyun Roh[5] rsh@urc.kr

[1]Automation and Systems Research Institute, Department of Electrical and Computer Engineering, Seoul National University,
[2]King's College London, London WC2R 2LS
[3]University of Strathclyde, Glasgow G1 1XH
[4]Department of Electronics Engineering, Sunchon National University
[5]REDONE TECHNOLOGIES CO., LTD

Abstract

Human-robot teaming is inevitable in various applications ranging from manufacturing to field robotics because of the advantages of adaptability and high flexibility. To become an effective team, knowledge regarding plan execution needs to be shared by verbalization. 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. In this paper, we introduce semantic scene understanding methods for verbalization of plan execution. We 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. Experiments were performed to verify the effectiveness of the proposed methods.

1 Introduction

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 has resulted in a considerable amount of attention being paid to such robots [WZG19]. 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 [TKL+14]. Cooperation between humans and heterogeneous robots can play an important role in adapting robots to an unstructured and dynamic environment [COGM19]. Many algorithms have been developed to resolve issues such as sensing, perception to planning, control, and safety in human-robot teaming [MdSB18, ZJSS17]. Among the various elements that need to be considered for a human-robot system, the most important function is verbalization of plan execution, which includes scene

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

understanding based on natural language as illustrated in Figure 1. This can enable humans and robots to share information in a form they can both understand, which is the most basic ability required for cooperation.

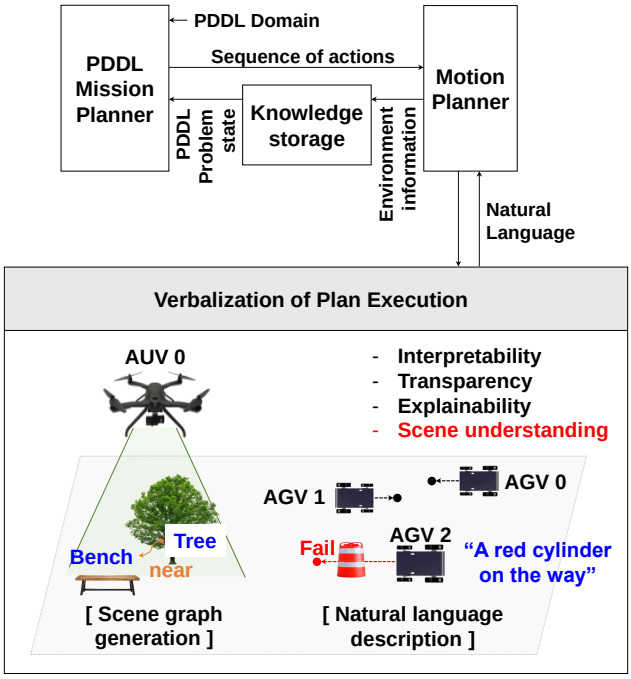


Figure 1: *Verbalization of plan execution, which is composed of interpretability, transparency, explainability and scene understanding plays an important role in human-robot teaming. In this paper, we focus on generating scene graphs and language descriptions for scene understanding.*

Semantic scene understanding is the process of perceiving environmental information in natural language or a form that can infer semantic meanings. In robotics, semantic mapping algorithms, which generate graphs that denote features and positions of detected objects as nodes, have been widely studied recently [ZWS+18, BADP17]. The graphs generated by these algorithms are unlike the maps generated by conventional methods, which consist of points, corners, lines, and planes. However, the generated semantic graph is rarely applied to data sharing methods for humans and robots, and these graphs need to be expressed in natural language. Natural-language-based scene understanding is studied in various forms such as image captioning [XBK+15, KFF15], visual question answering [GGH+17], and scene graph generation [WSW+17] in the field of computer vision. However, these methods are rarely applied to the semantic graph maps that are used by robots to represent the environment. Moreover, they do not address the problem of mission planning where humans and heterogeneous robots cooperate to achieve a common goal. In this paper, we generate scene graphs and language descriptions to focus on scene understanding, which is one fundamental element of verbalization of plan execution. A graph-based convolutional neural network [DBV16] is employed to generate sentences attention over graphs. An iterative message passing [XZCFF17] technique based on the gated recurrent unit (GRU) is used to generate scene graphs. We verified the proposed algorithms through experiments.

2 Approach

This section describes two methods of scene understanding using semantic graphs for plan execution verbalization. First, we generate natural language grounded scene graphs composed of objects as nodes, and their relationships as edges. Then, language descriptions that describe the overall scene are generated. The details are as follows.

2.1 Semantic graph generation

Semantic scene understanding based on graph maps is widely studied in robotics. However, these graph maps are rarely used for robotic applications such as mission planning, natural language processes, or plan execution ver-

balization. We address the issue of natural language-based surrounding scene understanding for the verbalization of plan execution using semantic graph maps. In this study, we assume that the graph map of the surrounding environment is generated in advance using semantic simultaneous localization and mapping (SLAM). To construct a similar graph with semantic SLAM, the features and position of objects are set as nodes. Features of objects are used for data association in the SLAM front-end. The position information of objects is utilized for graph optimization in the SLAM back-end. The generated semantic graph map G is illustrated in Fig 2. In this paper, multiple objects in the image are detected using a region proposal network [RHGS15] and encoded into feature vectors using the neural network, VGGNet [SZ14].The vector concatenated with the image information related to the $i-th$ object and the bounding box of the object is set to the feature vector f_i^v of node $v_i \in V$. We set the feature vector representing the union region of two objects as f_{ij}^e of edge $e_{ij} = (v_i, v_j) \in E_{ij}$ that connects v_i and v_j.

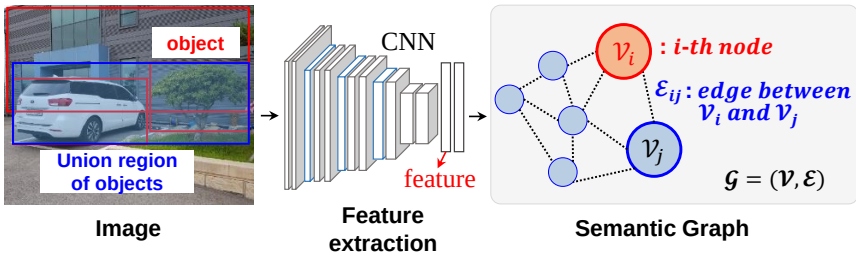


Figure 2: *Scene graph generation: Detected objects are encoded as graph node features. The union regions of two objects are encoded as graph edge features. A convolutional neural network is utilized for feature encoding.*

2.1.1 Graph inference

We infer the optimal word for each node and edge of the generated semantic graph. The graph inference process for the semantic graph f_i^v and f_{ij}^e is as follows.

$$g^* = argmax_g Pr(g \mid f_i^v, f_{ij}^e) \tag{1}$$

$$Pr(g \mid I, B_I) = \prod_{i \in V} \prod_{j \neq i} Pr(v_i^{class}, v_i^{bbox}, e_{ij} \mid f_i^v, f_{ij}^e) \tag{2}$$

where, C and R are a set of object classes and relationship types, $v_i^{class} \in C, v_i^{bbox} \in \mathbb{R}^4, e_{ij} \in R$. An iterative message passing model [XZCFF17] is utilized for graph inference. Node message pooling focuses on finding words for nodes through both the inbound and outbound edge states. Edge message pooling focuses on finding words for edges through both the object and subject states. Through repeated message pooling, we generate scene graphs comprising the most optimal words for each node and edge.

2.1.2 Language description

We generate language description for a semantic graph. The conventional methods utilizing convolutional neural network are rarely applied to graph data that is irregular and unstructured. We generate sentences using a graph convolutional neural network defined by spectral theory as follows.

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W_g^{(l)}) \tag{3}$$

where, $\hat{A} = A + I$ is an adjacency matrix A with self-connection I. $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ and $W_g^{(l)}$ are a degree matrix and a trainable variable, respectively. $H^{(l)} \in \mathbb{R}^{M \times D}$ is output of the l-th layer, where $H^{(0)} = X$. In this paper, a graph is encoded as a 1024-dimensional vector with a fully connected layer. Then, a concatenated vector composed of a graph feature and a word is fed into a recurrent neural network to predict the probabilistic distribution of words.

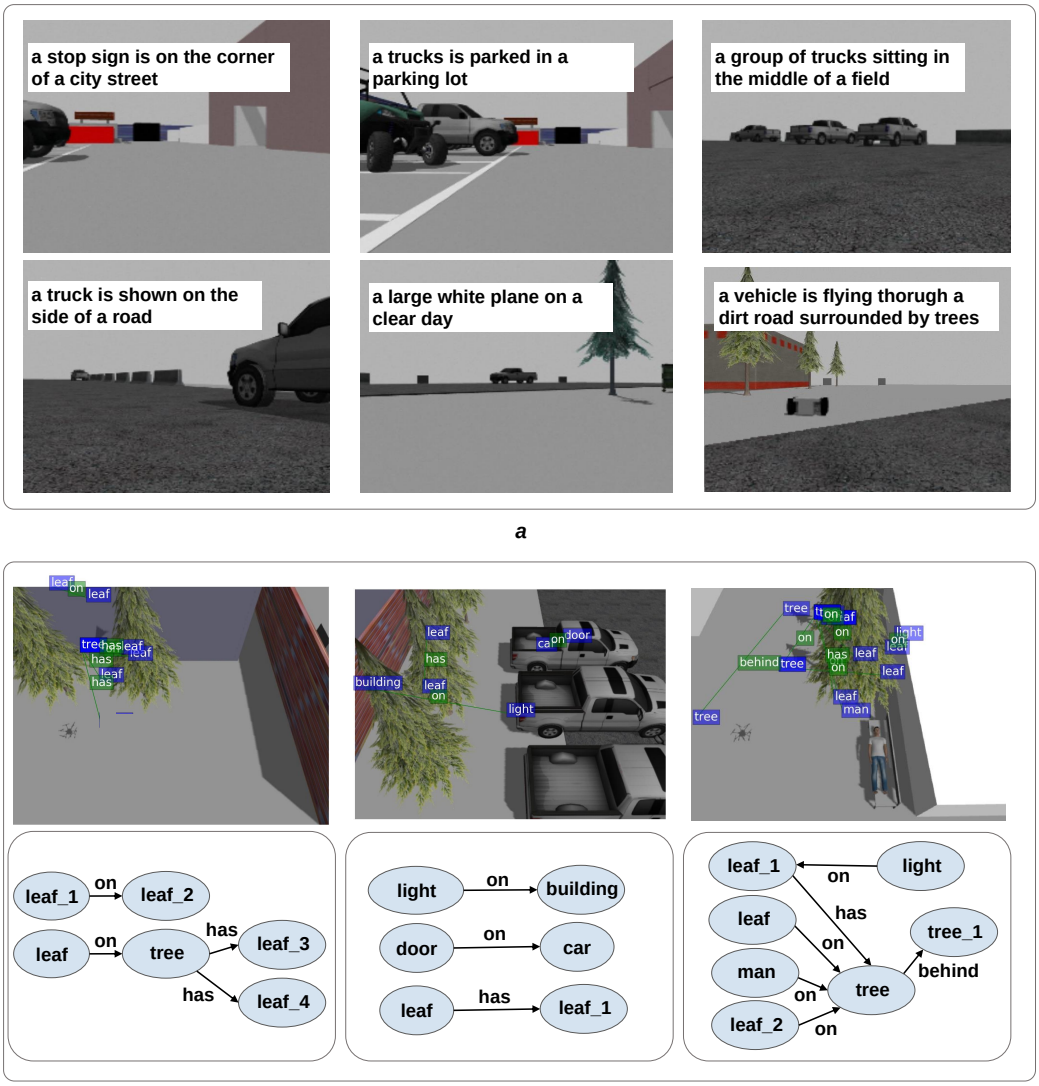


Figure 3: *Simulation Results: (a) Language description (b) Scene graph generation*

3 Experiment

We generated scene graphs and language descriptions using images of the surrounding environment obtained with mobile robots performing mission planning of surveillance in the simulation environment. Ubuntu 16.04, ROS Kinetic, and Gazebo 7 were used to set up the simulator. Two datasets were utilized for the neural network training. The network for language description was trained with a COCO dataset [LMB+14], whereas the network for scene graph generation was trained with a visual genome dataset [KZG+17]. The COCO dataset consists of images, object boundary boxes, and captions. The visual genome dataset is composed of annotations of object relationships and object labels. As these datasets only have images, we constructed graphs before training the networks; VGGNet [SZ14] was used for graph construction. We set the maximum number of nodes at 20 in order to cope with various sizes of graphs; when fewer than 20 nodes were present, empty nodes with zeros were added.

Even though the networks were trained with datasets from the real world, the proposed methods successfully generated language description and scene graphs for the simulation world as illustrated in Fig. 3. These results can be utilized for the verbalization of plan execution. Language description can contribute toward recovering from mission failure, as the failure of a robot is inevitable. For example, assume that a robot has to go to a

certain position and wait for a human to load a package. As it does so, a car blocks the path to the robot and the human cannot approach it; consequently, the robot will fail its mission. In this case, the robot can inform humans about the failure by describing the current situation and move to a new position to complete the mission. Scene graph generation can contribute toward gathering information in unseen and dynamic environments in a compact and communicable form. For example, assume that a robot is located in a place where a human cannot approach it. The generated scene graph can be used for humans to identify the place where the robot is located.

4 Artificial Intelligence Planning

AI Planning is a branch of AI 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 relationship between them. In particular, the model includes an initial state, possible actions that affect the state, as well as the desired goal condition.

A planner is used to find 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 in order 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. [CCFL10]) are equipped with various heuristics in order to find solutions faster than having to explore every state in the state space, thus enabling their use for planning and replanning online.

5 Planning and Plan Execution

Task planning for robots means planning with incomplete and unreliable data. Observations can be made from sensors in order to update the model used for planning and execution through state estimation. An up-to-date model for planning reduces the risk of plan failure, and can identify earlier when a plan under execution is no longer valid. However, even so it is likely that plans fail during execution, and in such cases it is critical that the robotic agent is able to explain to the operator exactly why.

The work presented in this paper can be usefully 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. Relations 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 enhance descriptions of the state that can be used to describe why the plan has failed. If a location has become unreachable because of an obstruction, a verbalization of the scene graph, such as the examples in Fig. 3, can be given to a operator as an explanation of plan failure. This allows the operator to understand how the environment is different from what was expected, and what to do next. In this section we discuss future work in this direction.

A team of robots can be controlled through task planning using the ROSPlan [CFL$^+$15] framework for task planning in ROS. The scene graph will be integrated with ROSPlan to perform continuous updates to the current state through an integration with the ROSPlan 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 a necessary function if the robot operates within a dynamic environment. Second, new objects that are detected can be immediately described in terms of their position and relation to other objects. This is a necessary step for the planner to understand how they can be used in a plan, or what effect they might have on the state.

Plan execution on board the robots will be extended to include verbalization describing the plan under execution. This will be done by integrating the verbalization component with the plan execution components of ROSPlan in the following two ways: first to provide verbalization of updates to the current state, and second to provide verbalization of obstructions that prevent the robot from achieving its goal. In human-robot teaming scenarios, it is important that the human operator is given sufficient situational awareness to judge 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 understand which unexpected event or object has resulting in the failure of the plan.

6 Conclusion

Verbalization of plan execution is the most fundamental component of human-robot collaboration in that it can share information in an interpretable form to achieve a shared goal. In this paper, two methods of semantic scene understanding are proposed for the verbalization of plan execution. A graph convolutional neural network and iterative message pooling are utilized to generate both language description and a scene graph, respectively. The proposed method was successfully verified with the simulator in our study.

Acknowledgement

This work was supported in part by Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Ministry of Trade, Industry & Energy (MOTIE) (No. 1415162366 and No. 141562820) and in part by a Bio–Mimetic Robot Research Center funded by Defense Acquisition Program Administration, and by Agency for Defense Development (UD190018ID).

References

[BADP17] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic slam. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1722–1729. IEEE, 2017.

[CCFL10] Amanda Coles, Andrew Coles, Maria Fox, and Derek Long. Forward-chaining partial-order planning. In *ICAPS*, pages 42–49, 2010.

[CFL+15] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtós, and M. Carreras. Rosplan: Planning in the robot operating system. In *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, 2015.

[COGM19] Micah Corah, Cormac O'Meadhra, Kshitij Goel, and Nathan Michael. Communication-efficient planning and mapping for multi-robot exploration in large environments. *IEEE Robotics and Automation Letters*, 4(2):1715–1721, 2019.

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[GGH+17] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5630–5639, 2017.

[KFF15] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[KZG+17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[MdSB18] Dannilo Samuel Silva Miranda, Luiz Edival de Souza, and Guilherme Sousa Bastos. A rosplan-based multi-robot navigation system. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 248–253. IEEE, 2018.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TKL+14] Stefanie Tellex, Ross A Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. In *Robotics: Science and systems*, volume 2, 2014.

[WSW+17] Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1367–1381, 2017.

[WZG19] Nana Wang, Yi Zeng, and Jie Geng. A brief review on safety strategies of physical human-robot interaction. In *ITM Web of Conferences*, volume 25, pages 1–3. EDP Sciences, 2019.

[XBK+15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[XZCFF17] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.

[ZJSS17] Shiqi Zhang, Yuqian Jiang, Guni Sharon, and Peter Stone. Multirobot symbolic planning under temporal uncertainty. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 501–510. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[ZWS+18] Liang Zhang, Leqi Wei, Peiyi Shen, Wei Wei, Guangming Zhu, and Juan Song. Semantic slam based on object detection and improved octomap. *IEEE Access*, 6:75545–75559, 2018.

Mental Simulation for Autonomous Learning and Planning Based on Triplet Ontological Semantic Model

Yuri Goncalves Rocha and Tae-Yong Kuc
College of Information and Communication Engineering, Sungkyunkwan University, South Korea
{yurirocha, tykuc}@skku.edu

Abstract

Cognitive science findings showed that humans are able to create simulated mental environments based on their episodic memory and use such environment for prospecting, planning, and learning. Such capabilities could enhance current robotic systems, allowing them predict the output of a plan before actually performing the action on the real world. It also allow robots to use this simulated world to learn new tasks and improve its current ones using Reinforcement Learning approaches. In this work, we propose a semantic modeling framework, which is able to express intrinsic semantic knowledge in order to better represent robots, places and objects, while also being a memory-efficient alternative to classic mapping solutions. We show that such data can be used to automatically generate a complete mental simulation allowing robots to simulate themselves and other modeled agents into known environments. This simulations allows robots to perform autonomous learning and planning without the need of human-tailored models.

1 Introduction

Mental simulation is one of the fundamental cognitive skills. It allow humans (and perhaps other animals too) to predict and anticipate outcomes by recalling past experiences. This ability is one of the main pillars of episodic memory [Boyer, 2008] paramount for task planning during navigation [Burgess, 2008]. Mental simulation theory presupposes three main components: behaviors can be simulated, perception systems can also be simulated, and, finally, outcomes can be anticipated by combining simulated behaviors and perception [Hesslow, 2012]. Early researches on the field also showed that the same mechanism is used to predict counterpart's thoughts and behaviors [Gordon, 1986]. Besides being one of the core mechanisms of the human brain, mental simulation is yet to be fully explored on robotic systems. Some works on the field suggested that such capability should not only be integrated into learning and planning algorithms, but also be their central architectural layer [Polceanu and Buche, 2017].

In order to simulate a given environment, it is paramount to understand it semantically. The semantic information adds another layer to the robot knowledge, allowing for a better understanding of the intrinsic concepts and relations that are inferred naturally by humans. Even though there are several applications of semantic data in the robotics field [Waibel et al., 2011, Kostavelis et al., 2016, Cosgun and Christensen, 2018], just a small fraction of them use it to perform mental simulations [Tenorth and Beetz, 2009, Beetz et al., 2015, Beetz et al., 2018].

Learning is one of the main applications of mental simulation. Humans, at first, learn by interacting with the environment and observing its output. After obtaining enough experience, the brain is able to simulate this environment, and use it to imagine new outcomes by applying a different behavior. In robotics, one of the main fields of Artificial Intelligence(AI) is Reinforcement Learning(RL). RL is inspired by the human way of learning, and it works by

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

exploring the environment and giving (or removing) rewards, depending on how well the robot executed a given task. More specifically, Deep Reinforcement Learning(deep-RL), has been used on several autonomous navigation applications [Tai et al., 2017, Shah et al., 2018, Kahn et al., 2018].

The contributions of this work are as follows:

- Expanding an Ontologic Semantic Framework in order to automatically generate a full simulation environment, including a simulated robot.
- An end-to-end deep-RL model for autonomous navigation trained using the mentally simulated environment.

2 Related Work

In the past decades, several works proposed ways to incorporate knowledge into computers. CYC [Lenat, 1995] and SUMO [Niles and Pease, 2001] gathered a large amount of encyclopedic knowledge into its database, however such knowledge lacked the information necessary for mobile robot tasks. The OMICS [Gupta et al., 2004] project created a similar database containing the necessary knowledge in order to a robot complete several indoor tasks. The RoboEarth [Waibel et al., 2011] project tried to create a World Wide Web for robots, where they would be able to share and obtain knowledge in an autonomous way. KnowRob [Tenorth and Beetz, 2009, Beetz et al., 2018] and OpenEASE [Beetz et al., 2015] created a complete knowledge processing system capable of semantic reasoning and planning, and also performing mental simulations (referred as *Mind's Eye*). Most of those works, however, focused on manipulation tasks only.

Despite being thoroughly studied by cognitive science researches [Boyer, 2008, Burgess, 2008, Hesslow, 2012, Kahneman and Tversky, 1981], the mental simulation concept only started to be applied to computational systems few decades ago. Most of the early works focused on the "putting yourself on other's shoes" approach, where an agent would simulate itself on its counterpart perceived state in order to infer about its feelings and intentions. Leonardo [Gray and Breazeal, 2005] was developed to infer a human intention and aid the execution of this predicted task. In [Buchsbaumm et al., 2005], an animated mouse was able to imitate similar actors by inference using its own motor and action representations. [Laird, 2001] created a Quake bot able to predict its opponent next action by simulating itself on the opponent's current state, while [Kennedy et al., 2009] used its own behavior model to predict another agent's actions. Most of the recent works on robotics field, however, focused on the application of mental simulation to manipulation tasks planning and learning [Tenorth and Beetz, 2009, Beetz et al., 2015, Beetz et al., 2018, Kunze and Beetz, 2017], or comprehension and expression of emotions when socializing with humans [De Carolis et al., 2017, Horii et al., 2016]. J. Hamrick [Hamrick, 2019], however, showed that there are several similarities between mental simulation findings from cognitive science and model-based deep-RL approaches.

Deep Reinforcement Learning (deep-RL) has been applied to several different robot tasks, including but not limited to Human-Robot Interaction [Christen et al., 2019, Qureshi et al., 2018], dexterous manipulation [Gu et al., 2017, Rajeswaran et al., 2017] and autonomous map-less navigation [Kahn et al., 2018, Zhu et al., 2017]. RL methods can be divided into *model-based* and *model-free value-based* approaches. Model-based algorithms, such as [Zhu et al., 2017] use a predictive function that receive the current state and a sequence of actions and outputs the future states. The policy then select the sequence of actions that maximizes the expected rewards from the predicted states. Model-free approaches, such as [Christen et al., 2019], approximate a function that receives the current state and action and outputs the sum of the expected future rewards. The policy then picks the action that maximizes this output. Generally, model-based approaches are sample-efficient, while model-free methods are better at learning complex, high-dimensional tasks. Some approaches [Qureshi et al., 2018, Kahn et al., 2018] also tried to use hybrid methods which would explore the advantages of both model-based and model-free value-based approaches. Regarding value-based deep-RL methods, Deep Q Network (DQN) has been vastly used by the research community [Qureshi et al., 2018], due to its good generalization capabilities and relatively simple training method. DQN, however, can only approximate a discrete action space, requiring continuous applications to be discretized beforehand. Trying to solve this issue, some new approaches such as Deep Deterministic Policy Gradient (DPPG) have been used [Christen et al., 2019, Gu et al., 2017] due to its ability to approximate continuous action spaces.

3 Triplet Ontological Semantic Model

Researches on the cognitive science and neuroscience fields [Burgess, 2008] have shown that the human brain has its own "GPS" mapping system. Every time we revisit a known environment this GPS is responsible for navigating using past known information and update itself with novel data. By relying on relational information instead of precise metric position, the human brain remains unparalleled on its spatial scalability and data efficiency. Robots, on the other hand,

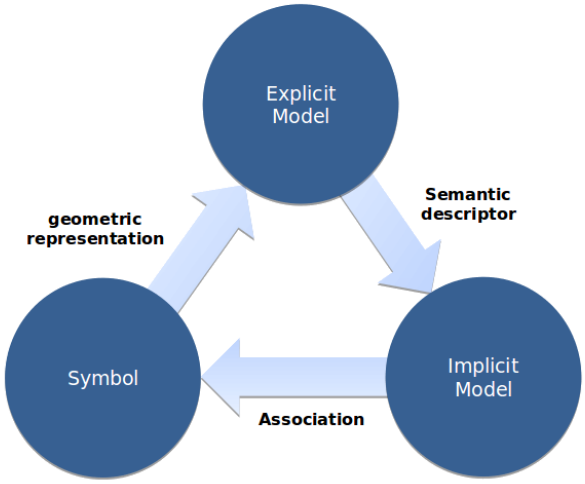


Figure 1: Triplet Ontologic Semantic Model (TOSM) representation.

still heavily rely on information-rich, yet memory inefficient, maps in order to localize themselves and navigate through known environments. Despite being precise, those maps require a large amount of data to be stored, which hinders the robot's long-term autonomy on large scale environments due to lack of storage space. Aiming to mimic the brain GPS model efficiency, the Triplet Ontological Semantic Model(TOSM) [Joo et al., 2019] was developed.

The TOSM representation can be described as three interconnected models as shown on Fig. 1. The explicit model subsumes everything that can be measured or obtained through sensorial means. It can be data such as size, three-dimensional pose, shape, color, texture, etc, which are already vastly used on current robot applications. The implicit model, on the other hand, contains intrinsic knowledge which cannot be obtained by sensors alone, thus needing to be inferred from the available semantic information. The implicit model comprise a large variety of data that range from physical properties such as mass and friction coefficient, relational data (e.g. object A is inside object B), until more complex semantic information such as "An automatic door opens if one waits in front of it". Finally, the symbolic model describes an element using a language-oriented way, such as name, description, identification number and symbols that can represent such element.

By creating an environment database using TOSM encoded data, a hierarchical mapping system was created, based on the findings of cognitive science. As shown on Fig. 2, different maps can be generated on-demand according to the specifications of the robot and the given task. This eliminates the demand to store several different maps by being able to build them only when needed, reducing the data redundancy and improving its storage efficiency. The TOSM can be also used to model places and robots, which combined with the object models can be used to generate high-level semantic maps.

In this work, we also used the TOSM encoded on-demand database to automatically generate a complete simulation environment without the need of domain expert tailored models. This allows the robot to update its mental simulated world automatically just by updating the on-demand database. In order to encode the TOSM data into a machine-readable format, the Ontology Web Language (OWL) was used. OWL is widely used and has an active community which created several tools and applications openly available. We used one of those tools, the Protégé framework, to manipulate and visualize the OWL triplets.

3.1 Robot Description

In order to describe a robot, it was divided into structural parts, sensors, wheels and joints, each of them described by its own explicit, implicit and symbolic information. All categories contain similar explicit data, such as pose, shape, color, size and material. The symbol data contains the part name and an identification number. On the other hand, the implicit data is unique for each category. For structural parts, it contains the mass and the material, while wheels also store whether or not it is a active wheel. Joints store which two parts it is connected to. Moreover, the implicit information can be different for each type of sensor. For example, cameras were described by image resolution, field of view, frames per second and, for RGB-D cameras, range. A laser range finder can have data such as range, view angle and number of samples.

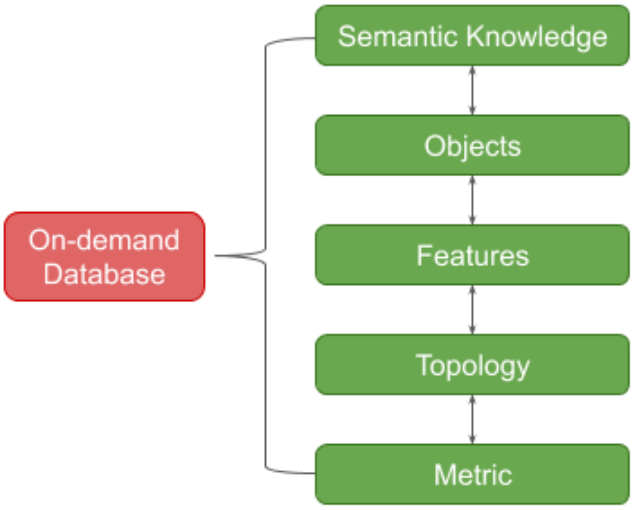


Figure 2: On-demand map generation.

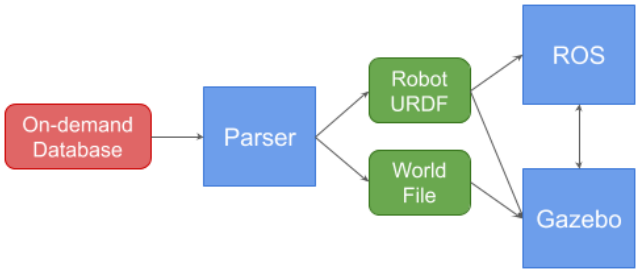


Figure 3: Data flow for the mental simulation.

3.2 Environment Description

The environment can be modeled in a similar fashion as the robot. It is divided mainly into objects and places. Regarding objects, the explicit model contains the same data as described on Subsection 3.1. The implicit model contains data such as mass, material and relational spatial information, such as ”in front of”, ”on the left of”, etc. With respect to places, on the other hand, the explicit model contains its boundary points, while the implicit model stores which objects/places are inside of it and which other places is it connected to. The symbol information is the same for both, storing the name of the place/object and an identification number.

4 Mental Simulation

By encoding the TOSM information using the OWL format, it is possible to do semantic reasoning and querying. Before doing any task, the robot can reason about its feasibility by knowing about its surrounding environment’s characteristics and its own structure, limitations and properties. For example, a robot only equipped with a laser scanner can reason about its inability to navigate through a corridor made out of glass walls. We extended those reasoning capabilities by automatically generating a complete mental simulation environment using only the on-demand database data.

The data flow for the mental simulation can be seen on Fig 3. Whenever its needed, the robot requests the TOSM data to the on-demand database, and generate two different outputs. The first one is an Universal Robot Description Format(URDF) which is then fed into Robot Operating System(ROS) and Gazebo Simulator in order to control and simulate the virtual robot. The second one is a Gazebo World file which represents the whole environment simulation. Those files are generated on-demand and can be constantly updated whenever the real robot update its database.

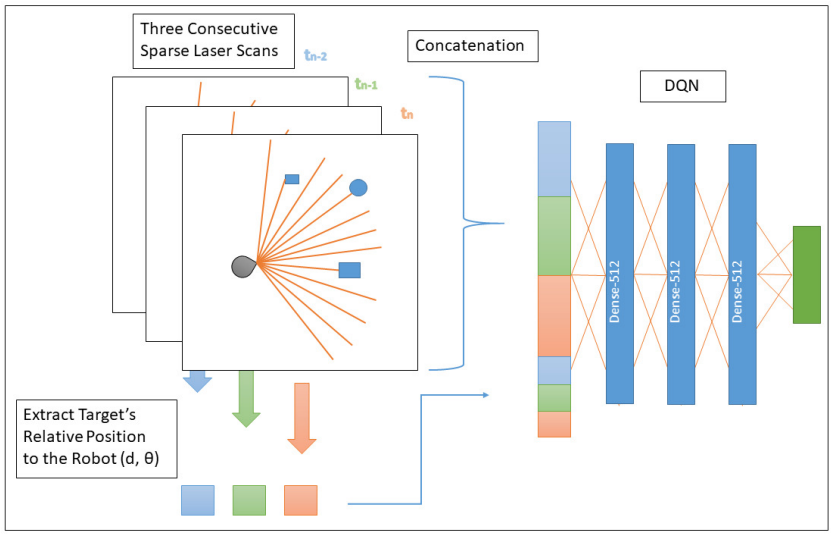


Figure 4: DQN structure.

5 Reinforcement Learning for Autonomous Navigation

In order to show one os the uses for the mental simulator, an autonomous navigation policy was trained using a DQN. The training was performed using a Core i7 CPU and a Nvidia GTX 1060. The OpenAI ROS framework [ezq,] was used in order to to abstract the layer between the reinforcement learning algorithm and the Gazebo/ROS structure. The task learning architecture is shown on Fig. 4. The observation space is composed by the latest three sparse laser scans concatenated with the last three relative distances between the robot and the target way-point. The action space are 15 different angular velocities equally distributed from $-0.5rad/s$ to $0.5rad/s$. The rewards were defined as

$$\begin{cases} r_{completion}, & \text{if at goal,} \\ r_{closer}, & \text{if getting closer to the goal,} \\ r_{collision}, & \text{if too close to an obstacle,} \end{cases}$$

where $r_{completion}$, r_{closer} and $r_{collision}$ were defined trivially.

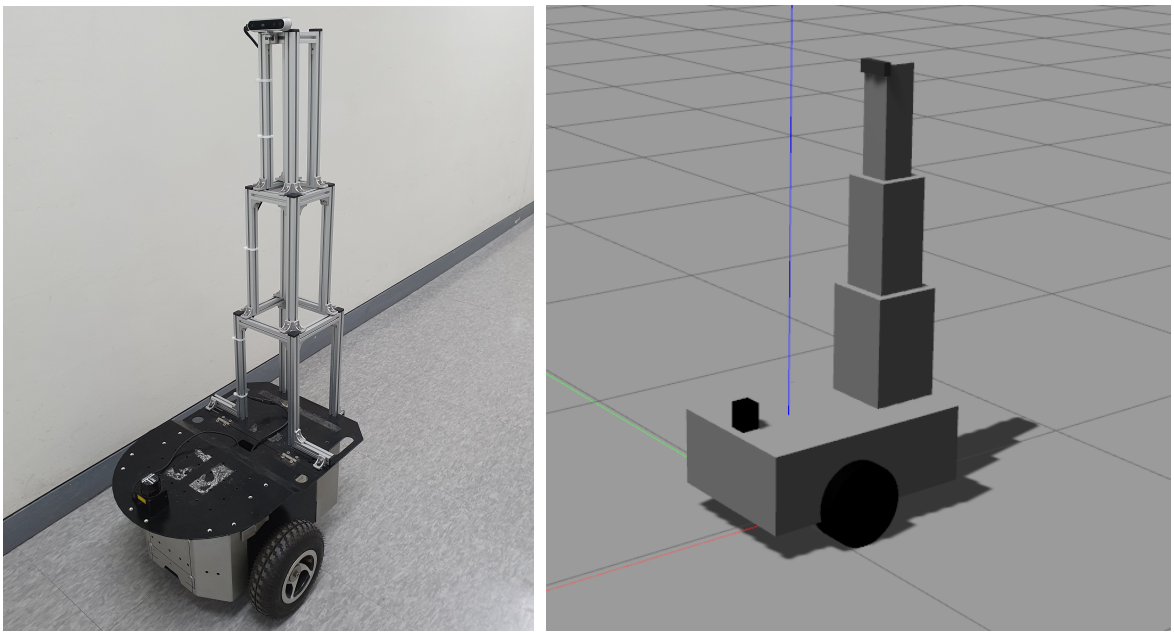
The training was done using an ε-greedy exploration approach, where ε started at 1.0 and decayed until 0.1. The DQN was trained using batches of 64, with learning rate $\alpha = 0.001$ and discount factor $\gamma = 0.996$. The robot was trained for a total of 2000 episodes, where each episode would end in case of completion, collision or after 1000 steps.

6 Results and Discussion

In order to show the usability of such framework, the 7th floor of the Corporate Collaboration Center, Sungkyunkwan University, was added to the on-demand database. Additionally, the differential robot shown on Fig. 5a was modeled. The comparison between the simulated world and the database data can be seen on Fig. 6 while the comparison between the real and simulated robot is shown on Fig. 5.

By automatically generating this simulation environment, we allow the robot to perform mental simulation without the aid of domain experts by reusing the same data it already uses for planning and navigation. Such approach further improves the robot autonomous behavior by letting it simulate itself(or even other robots) on *its own mind* and use this simulated environment to prospect about new actions. Currently, this can be done in two different ways:

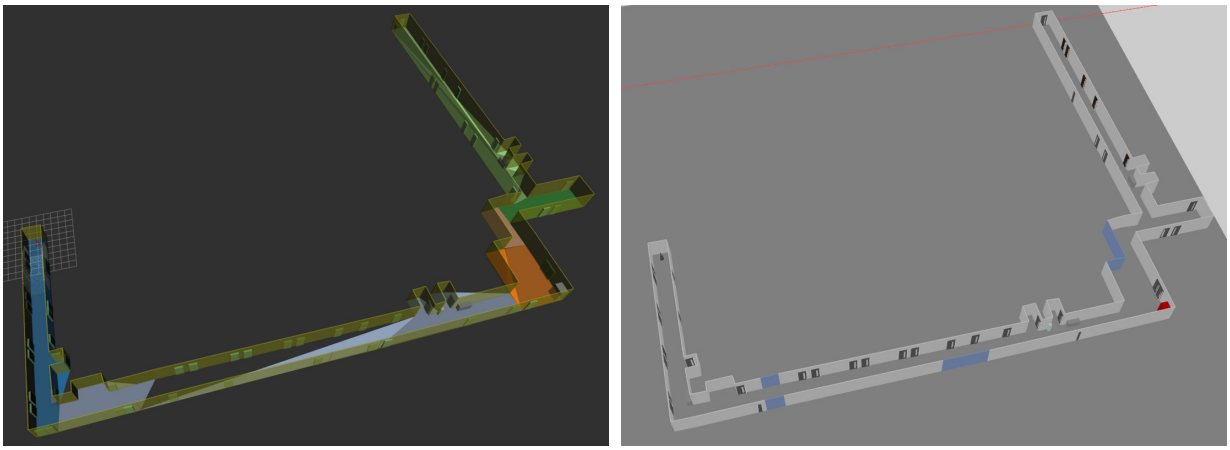
- **Learning**: The robot can use the mental simulation to run reinforcement learning algorithms in order to train and learn the execution of new tasks. This is mainly done when the physical robot is idle(e.g. charging at night).
- **Planning**: The robot can simulate its current state and use it for testing a plan, generated by traditional planners, and check whether it succeeds or not. In the case of failure, the robot can re-plan without having to fail on the real environment, allowing for a more robust task execution.



(a) Real robot

(b) Simulated robot

Figure 5: Real and simulated robots comparison.



(a) Visualization of the data obtained from the on-demand DB. Objects are represented as bounding boxes, while places are represented as colored polygons on the floor

(b) Mental simulation environment

Figure 6: Comparison between environment data queried from the DB and mental simulation.

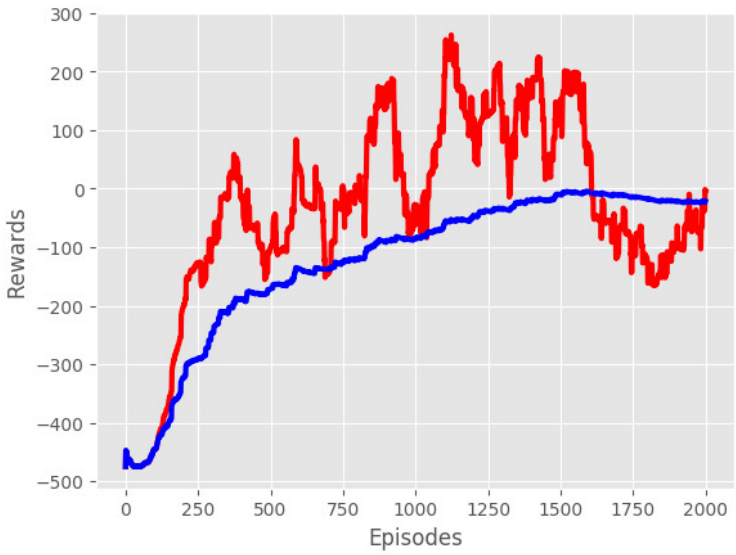


Figure 7: The blue line represents the cumulative average reward, while the red line shows the 100 episodes moving average.

The main advantage of such simulation is that it removes the necessity of a *tailor-made* simulation environment, allowing the robot to generate and update this environment automatically. It can be specially useful for reinforcement learning approaches, which, in theory, gets better the more experiences the robot collects. The robot should be able to run learning algorithms whenever it is idle, slowly improving itself. Naturally, a cluster running multiples CPUs and GPUs would learn orders of magnitude faster, allowing the robot to run the learning algorithms itself bring the robotics field one step closer to true robot autonomy. Finally, by uploading the on-demand database to a cloud infrastructure, robots should be able to share its own model and environment maps, allowing other robots to compare its performance on a given task with one another, and provide this information for its operators automatically.

By using the mental simulation, a autonomous navigation task was learned. The average reward graph can be seen on Fig. 7. Despite being one of the simpler deep-RL approaches, DQN was shown to be good at generalizing a high-dimensional task. However, the whole training took around 30 hours on a mid-range computer. If the same training were performed on a mobile robot, the training times might be too prohibitive. Thus, sample-efficient learning algorithms should be more appropriate for this application.

7 Conclusion and Future Work

In this paper, we presented a method of generating an automatic mental simulation by using a TOSM on-demand database. By allowing robots to create and update mental simulations on a complete autonomous way, we removed the necessity of expert-tailored models, leading for more autonomous robotic systems. In order to show one of the possible applications of such method, we trained the robot to autonomously navigate on an known environment by using a Deep Q Network. We plan now to expand those applications, by including behaviors into the on-demand DB, allowing robots to share and configure RL policies by themselves. We also want to explore the usability of our framework when combined with classical planners.

Acknowledgment

This research was supported by Korea Evaluation Institute of Industrial Technology(KEIT) funded by the Ministry of Trade, Industry & Energy (MOTIE) (No. 1415162366 and No. 1415162820)

References

[ezq,] Openai ros documentation. Date last accessed 04-Aug-2019.

[Beetz et al., 2018] Beetz, M., Beßler, D., Haidu, A., Pomarlan, M., Bozcuoğlu, A. K., and Bartels, G. (2018). Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE.

[Beetz et al., 2015] Beetz, M., Tenorth, M., and Winkler, J. (2015). Open-ease. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1983–1990. IEEE.

[Boyer, 2008] Boyer, P. (2008). Evolutionary economics of mental time travel? *Trends in cognitive sciences*, 12(6):219–224.

[Buchsbaumm et al., 2005] Buchsbaumm, D., Blumberg, B., Breazeal, C., and Meltzoff, A. N. (2005). A simulation-theory inspired social learning system for interactive characters. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 85–90. IEEE.

[Burgess, 2008] Burgess, N. (2008). Spatial cognition and the brain. *Annals of the New York Academy of Sciences*, 1124(1):77–97.

[Christen et al., 2019] Christen, S., Stevsic, S., and Hilliges, O. (2019). Guided deep reinforcement learning of control policies for dexterous human-robot interaction. *arXiv preprint arXiv:1906.11695*.

[Cosgun and Christensen, 2018] Cosgun, A. and Christensen, H. I. (2018). Context-aware robot navigation using interactively built semantic maps. *Paladyn, Journal of Behavioral Robotics*, 9(1):254–276.

[De Carolis et al., 2017] De Carolis, B., Ferilli, S., and Palestra, G. (2017). Simulating empathic behavior in a social assistive robot. *Multimedia Tools and Applications*, 76(4):5073–5094.

[Gordon, 1986] Gordon, R. M. (1986). Folk psychology as simulation. *Mind & Language*, 1(2):158–171.

[Gray and Breazeal, 2005] Gray, J. and Breazeal, C. (2005). Toward helpful robot teammates: A simulation-theoretic approach for inferring mental states of others. In *Proceedings of the AAAI 2005 workshop on modular construction of human-like intelligence*.

[Gu et al., 2017] Gu, S., Holly, E., Lillicrap, T., and Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE.

[Gupta et al., 2004] Gupta, R., Kochenderfer, M. J., Mcguinness, D., and Ferguson, G. (2004). Common sense data acquisition for indoor mobile robots. In *AAAI*, pages 605–610.

[Hamrick, 2019] Hamrick, J. B. (2019). Analogues of mental simulation and imagination in deep learning. *Current Opinion in Behavioral Sciences*, 29:8–16.

[Hesslow, 2012] Hesslow, G. (2012). The current status of the simulation theory of cognition. *Brain research*, 1428:71–79.

[Horii et al., 2016] Horii, T., Nagai, Y., and Asada, M. (2016). Imitation of human expressions based on emotion estimation by mental simulation. *Paladyn, Journal of Behavioral Robotics*, 7(1).

[Joo et al., 2019] Joo, S.-H., Manzoor, S., Rocha, Y. G., Lee, H.-U., and Kuc, T.-Y. (2019). A realtime autonomous robot navigation framework for human like high-level interaction and task planning in global dynamic environment. *arXiv preprint arXiv:1905.12942*.

[Kahn et al., 2018] Kahn, G., Villaflor, A., Ding, B., Abbeel, P., and Levine, S. (2018). Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE.

[Kahneman and Tversky, 1981] Kahneman, D. and Tversky, A. (1981). The simulation heuristic. Technical report, STANFORD UNIV CA DEPT OF PSYCHOLOGY.

[Kennedy et al., 2009] Kennedy, W. G., Bugajska, M. D., Harrison, A. M., and Trafton, J. G. (2009). "like-me" simulation as an effective and cognitively plausible basis for social robotics. *International Journal of Social Robotics*, 1(2):181–194.

[Kostavelis et al., 2016] Kostavelis, I., Charalampous, K., Gasteratos, A., and Tsotsos, J. K. (2016). Robot navigation via spatial and temporal coherent semantic maps. *Engineering Applications of Artificial Intelligence*, 48:173–187.

[Kunze and Beetz, 2017] Kunze, L. and Beetz, M. (2017). Envisioning the qualitative effects of robot manipulation actions using simulation-based projections. *Artificial Intelligence*, 247:352–380.

[Laird, 2001] Laird, J. E. (2001). It knows what you're going to do: adding anticipation to a quakebot. In *Proceedings of the fifth international conference on Autonomous agents*, pages 385–392. ACM.

[Lenat, 1995] Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

[Niles and Pease, 2001] Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM.

[Polceanu and Buche, 2017] Polceanu, M. and Buche, C. (2017). Computational mental simulation: A review. *Computer Animation and Virtual Worlds*, 28(5):e1732.

[Qureshi et al., 2018] Qureshi, A. H., Nakamura, Y., Yoshikawa, Y., and Ishiguro, H. (2018). Intrinsically motivated reinforcement learning for human–robot interaction in the real-world. *Neural Networks*, 107:23–33.

[Rajeswaran et al., 2017] Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. (2017). Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*.

[Shah et al., 2018] Shah, P., Fiser, M., Faust, A., Kew, J. C., and Hakkani-Tur, D. (2018). Follownet: Robot navigation by following natural language directions with deep reinforcement learning. *arXiv preprint arXiv:1805.06150*.

[Tai et al., 2017] Tai, L., Paolo, G., and Liu, M. (2017). Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36. IEEE.

[Tenorth and Beetz, 2009] Tenorth, M. and Beetz, M. (2009). Knowrob—knowledge processing for autonomous personal robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4261–4266. IEEE.

[Waibel et al., 2011] Waibel, M., Beetz, M., Civera, J., d'Andrea, R., Elfring, J., Galvez-Lopez, D., Häussermann, K., Janssen, R., Montiel, J., Perzylo, A., et al. (2011). Roboearth-a world wide web for robots. *IEEE Robotics and Automation Magazine (RAM), Special Issue Towards a WWW for Robots*, 18(2):69–82.

[Zhu et al., 2017] Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE.

Combining Semantic Modeling and Deep Reinforcement Learning for Autonomous Agents in Minecraft

Andrew Melnik, Lennart Bramlage, Hendric Voss, Federico Rossetto, Helge Ritter
CITEC, Bielefeld University
33619 Bielefeld, Germany
andrew.melnik.papers@gmail.com

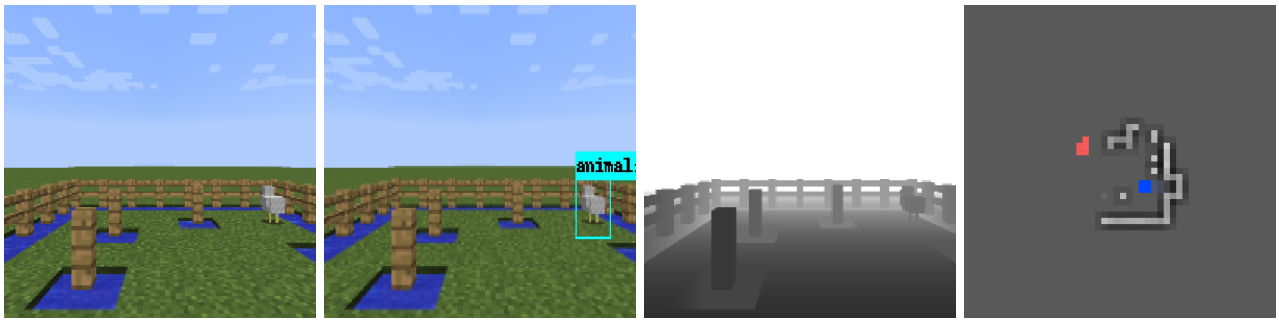


Figure 1: RGB observation, object detection, depth estimation, top-down map reconstruction.

1 Introduction

This work describes our extended solution [MARb] for the MARLÖ (Multi-Agent Reinforcement Learning in MalmÖ) Minecraft competition [MARa, PLHM+19, JHHM, GCH+19], and provides insights about how a semantic model of abstract representations of rich sensory data in an environment influence learning and performance in a task. Reinforcement learning can achieve state-of-the-art performance on a broad range of tasks [KMO+18, SM18], however, facing limitations in generalization and sample efficiency. Abstract latent representations and different levels of control can substantially increase sample efficiency [MFSR19]. Learning of a semantic model requires a structured representation of an environment. For that, a preprocessing pipeline can translate rich sensory information into useful abstractions and an object-based representation. A trained model can be utilized to provide navigational goals to the lower-level controller trained with Hindsight Experience Replay (HER) [AWR+17]. Universal value function approximators (UVFAs) and HER [AWR+17] allow efficient learning when a goal is provided.

2 Methods

To facilitate semantic inference we borrow from methods successfully employed in robotics and autonomous driving, namely U-Net [RPB15], SLAM, OctoMap [HWB+13] and an object detection module [Goo19]. This allows the agent to consider the locations and spatial relationships of all elements of the environment based solely on RGB-image data and the agent's actions. In the MARLÖ environment, any action is elementary and performed in exclusion of any other action (move one cell forward, turn left or right 90 degrees). We trained an open-source Keras implementation of U-Net [RPB15] on 841322 pairs of RGB and ground-truth depth images collected from the Minecraft simulator. We used Tensorflow Object Detection Library [Goo19] for tracking the second player, pet and exits in the challenge. Training data was collected in different game sessions by manually labeling 10000 frames. Combined with our action-based SLAM approach, continuous integration of new distance data from the depth estimator results in robust scene reconstructions as 3D Octomap [HWB+13] voxel volumes.

Copyright © 2019 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Table 1: Predefined concepts.

Entity	Relation	Entity
Object B	is attached (adjacent) to	Object A
Object B	is on the left side of	Object A
Object B	is on the right side of	Object A
Object B	is above	Object A
Object B	is below	Object A
Object B	is on the same vertical line as	Object A
Object B	is on the same horizontal line as	Object A

The Octomap is queried every step in the game environment to update the abstract top-down map representation (Fig. 1).

We consider a mixed reinforcement learning and semantic learning framework with a pre-processing pipeline. The pipeline translates RGB-images $s_t \in S$ into an abstract representation $z_t \in Z$ of a top-down map suitable for navigation (Fig.1), and tracks objects $o \in O$ with the object recognition module. We translate the abstract representations $z_t \in Z$ into a Boolean space of concepts $c_t \in C$ and determine causality of rewarding events in $C \in B^c$ (c is the number of concepts). A concept checks for a set of relationships of objects in the environment and provides a boolean test result. Along with the test result, each concept collects a relative position of the tested object. We predefined a set of object-related spatial concepts which are computed for each pair of objects in the "CatchTheMob" environment at each time step (Table 1). We used human-players demonstrations (20 episodes) to collect trajectories ($s_t \in S$), actions, and rewards. The objective is to leverage human-player demonstrations to acquire a semantic model of rewarding states that enables the agent to select navigational goals in the environment. When a set of rewarding concepts is selected, their superposition determines a distribution of positions related to the reward. We feed a sample from this distribution as a goal for the policy that controls the agent. The policy is trained with HER [AWR+17] to navigate the agent to a goal position in the top-down map ($z_t \in Z$).

3 Untangling causality

Here we provide experimental results which highlight the structure of the problem in the "CatchTheMob" environment [MARa]. We trained a DQN using the Rainbow implementation [HMVH+18] to catch the pet with two agents (black curve in Fig. 2) in the "CatchTheMob" environment [MARa] on the reconstructed top-down map representations (Fig. 1). The reward function returns +1 when the two agents are at adjacent positions to the pet but from two opposite sides. The rewarding condition of being at two opposite sides of the pet can be learned end-to-end or resolved by a higher-level planning model. To measure the potential benefit of the model we trained the same DQN [HMVH+18] in a simplified single-agent "CatchTheMob" environment. The reward function returns +1 when a single agent is at a position adjacent to the pet (blue curve in Fig. 2). We got more than 50 times faster convergence than in the two-agent case. The red curve shows training of a single player with

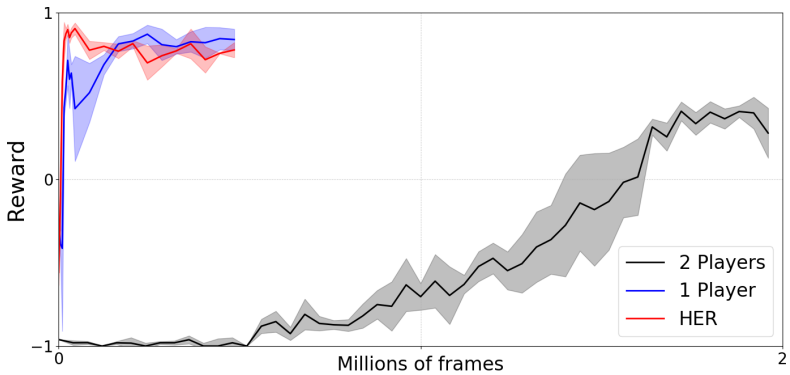


Figure 2: Convergence curves of agents trained on a 10x10 grid map. OY: cumulative reward (median of three seeds). The black curve shows training with two players, blue and red - with one player. In all experiments the episode runs until the reward +1 or the maximum of 50 steps is reached, the reward function gives -0.02 for each non-terminal action.

HER [AWR+17] to navigate to a given position. We got similar results to the single-agent case. These results highlight the benefit of higher-level planning through learning causality in the environment.

To learn the rewarding causality we select concepts activated at the rewarding events ($c_t \rightarrow c_{t+1}$, $r == 1$) as candidates for the necessary (but not sufficient) condition. Necessary concepts are encapsulated into a context and therefore not sufficient to fully determine the rewarding causality. To determine the sufficient set of concepts we optimize search by prioritizing concepts with a difference in rewarding / not rewarding samples and y a localization loss (1).

$$Loss = a\mathbf{P}(C_x) + b\mathbf{MSE}(D) \tag{1}$$

$\mathbf{P}(C)$ - reward prediction error of the selected set x of concepts C.
$\mathbf{MSE}(D)$ - mean square error of distribution of possible rewarding positions D. a, b - coefficients.

References

[AWR+17] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

[GCH+19] William H Guss, Cayden Codel, Katja Hofmann, Brandon Houghton, Noboru Kuno, Stephanie Milani, Sharada Mohanty, Diego Perez Liebana, Ruslan Salakhutdinov, Nicholay Topin, et al. The minerl competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:1904.10079*, 2019.

[Goo19] Google Brain. Tensorflow object detection api. `https://github.com/tensorflow/models/tree/master/research/object_detection`, 2019. Accessed: 2019-09-10.

[HMVH+18] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[HWB+13] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[JHHM] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell Microsoft. The Malmo Platform for Artificial Intelligence Experimentation *. Technical report.

[KMO+18] Łukasz Kidziński, Sharada Prasanna Mohanty, Carmichael F Ong, Zhewei Huang, Shuchang Zhou, Anton Pechenko, Adam Stelmaszczyk, Piotr Jarosik, Mikhail Pavlov, Sergey Kolesnikov, et al. Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 121–153. Springer, 2018.

[MARa] Crowdai marlo: Multi-agent reinforcement learning in minecraft. `https://www.crowdai.org/challenges/marlo-2018`.

[MARb] Github - marlo solution. `http://rebrand.ly/GitHub-MARLO`.

[MFSR19] Andrew Melnik, Sascha Fleer, Malte Schilling, and Helge Ritter. Modularization of end-to-end learning: Case study in arcade games. *arXiv preprint arXiv:1901.09895*, 2019.

[PLHM+19] Diego Perez-Liebana, Katja Hofmann, Sharada Prasanna Mohanty, Noburu Kuno, Andre Kramer, Sam Devlin, Raluca D. Gaina, and Daniel Ionita. The Multi-Agent Reinforcement Learning in Malm\"O (MARL\"O) Competition. jan 2019.

[RPB15] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[SM18] Malte Schilling and Andrew Melnik. An approach to hierarchical deep reinforcement learning for a decentralized walking control architecture. In *Biologically Inspired Cognitive Architectures Meeting*, pages 272–282. Springer, 2018.

How Does a Robot Speak? About the Man-Machine Verbal Interaction

Pierre-André BUVET[1], Bertrand FACHE [2] and Abdelhadi ROUAM[3]

[1] Paris 13 University, Théories Textes Numérique, 99 Avenue Jean Baptiste Clément, 93430 Villetaneuse, France
pierre-andre.buvet@univ-paris13.fr
[2] TEAMNET, 10 Rue Mercœur, 75011 Paris, France
b.fache@teamnet.fr
[3] ONTOMANTICS, 959 Rue de la Bergeresse, 45160 Olivet, France
abdelhadi.rouam@ontomantics.com

Abstract. We present here an intelligent man-machine interface which will be integrated to social robots dedicated to the elderly. Firstly, we discuss the specificities of conversational agents, also called chatbots, by comparing them to the other man-machine interfaces. Then, we establish how linguistic intelligence contributes to the development of high quality conversational agents. We finish by explaining the functioning of a conversational agent based on a linguistic processing of non-structured information.

1 Introduction

We are witnessing the advent of robots in our society, as Issac Asimov the writer of science fiction predicted in the 50s [1]. If the existing robots are way behind to be considered as well performing as the ones in the science fiction literature, huge and remarkable progresses have been made in the field of robotics in the last few years, especially robots functioning as caregivers 'social robots'. One of the characteristics of these robots is the integration of a system of verbal interactions between men and machines. Hence, this article highlights the characteristic of man-machine verbal interactions.

The senior market in rich countries is striking due to the increase rate of aging of the population which is linked to the growth of life expectancy. The need to accompany this phenomenon by practices, so called, aging well is a huge stake for these societies. As seniors advance in age, the question of losing their autonomy raises. The projections show an increasing gap between the non-autonomous people and the professionals who help them overcome their difficulties. Therefore, robots are presented as a sustainable solution to compensate the lack of staff [2].

Robots, which give people a helpful hand, are characterized by their aspect, mobility and their communicative capacity. The first characteristic is the aspect which is more or less humanoid. Besides NADINE, a social robot which resembles exactly to its designer Nadia Thalman, other robots have a general shape which is similar to the human body. It is just their sizes which look more like an infant than an adult, for example, the robots PEPPER, NAO and ROMEO of SoftBanks Robotics Company. Some robots have a more sophisticated shape which bears little resemblance to the human body; such as, the ELLI-Q Company robots. Sometimes, their upper part looks like a human head, for example, the robots CUTTI or the robot KURI of Mayfield Robotics Company. There exists, also, robots which have the appearance of an animal, either totally as the seal robot PARO of Inno3Med company, or partially as the bear robot ROBEAR of Riken company. The second characteristic concerns the mobility or the absence of mobility of the robot. The most humanoid robots have a manner of movement which simulates human walking, for example, the version ZORA of NAO. Less humanoid robots use wheels to move,

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

for example the robots of Kompaï Company or the robot GIRAFF of GiraffPlus consortium. Other robots have mobile elements but they do not move, for example the robots ORIGIBOT of Origin Robotics Company. The third characteristic is the man-machine interaction. It can be focused on the non-verbal communication (tactile, auditory or kinesic), in order to create emotions for users with cognitive difficulties, which is the case with the therapeutic robot PARO. The use of a screen integrated in the robot is another possibility, for example the range of robots of Padbot Company. The interaction with a screen may be tactile or vocal. At last, robots allow the verbal interactions through answering questions and obeying orders. Thus, the version ZORA of NAO interacts with its users through conversations.

Robots, which give a helpful hand to old individuals, have many functions which mobilize specific technologies. The function of care is the oldest one. It consists of measuring the physiological state of a person by pressing on objects connected to a platform dedicated to medical supervision. When the signals detect abnormalities, an alert is launched. Another function is the prevention of individuals' falls. It is applicable to robots which can be called smart walkers. They help people with balance troubles to move without risks. Developed computer systems manage the movements of these robots on the basis of the users' indications and information about the spatial environment obtained from sensors. Another function of robots is the detection of falls. They, also focus on the geo-location of sensors. They inform a system which measures the position of the user and launches an alert when this position is out of the security norms. Tele-presence is a broader function. On the one hand, it allows watching after a senior from a distance by moving the robot in the senior's environment and launching an alert in case of a problem. On the other hand, it communicates with the users. It involves a screen, which uses the videoconference and a system of remote control from a distance to move the screen mount. The function of sociability can be the fruit of tele-presence since it includes interactions between the user and another person. More broadly, the function of sociability concerns the companion robots, which are developed to compensate the emotional and social isolation of elderly. The function of helpfulness is consubstantial with robots since they perform tasks they were designed for. The last two functions, among others, focus on the technologies of chatbot type when it comes to verbal interactions. The quality of the chatbot is a crucial factor for the functions to work effectively.

For now, we are discussing only about chatbots. After placing the conversational agents[1] at the man-machine interface, we precise the contribution of linguistic intelligence to the development of an elaborated chatbot, then we explain how a conversational agent, based on a linguistic processing of a non-structured information, functions.

2 About the conversational agents

Digital technologies process the human language with its written and oral forms. These digital technologies either interpret it or automatically reproduce it. They are dedicated either exclusively to the first part of processing, for example the analysis of emotions, or specifically to the second part, for example automated journalism, or to both parts simultaneously. Digital technologies dedicated to the simultaneous processing of both parts are regularly man-machine interfaces known as 'chatbot' or 'conversational agents'. Chatbot, which is a compound English word, consists of the word chat and robot. A chatbot is presented in the form of an API which takes, as input, a sound or a text file and produces, as output, a sound or text file. This is followed by means of communication with the machine either in writing or speaking. The chatbot functions through the understanding of a message sent by a human being, a request, and through the formulation of another message, linked to the initial one, by the machine, a response.

Forms, interactive menus, command prompts, etc. allow the user to give instructions to a computer. Such modes of communication are, sometimes, insufficient to meet the user's exact needs due to their rigid relativity.

[1] The terms conversational agent and chatbot are synonyms.

The man-machine dialogue is conceived as a means to deal with these inadequacies. It consists of a system that allows an interaction, which is less or more narrow, between a human and a system. Its implementation is difficult because of the complexity of the language.

Therefore, the man-machine dialogue is a man-machine interface type as the language is a means to exchange the data. The word *dialogue*[2] indicates " different shapes of interviews between two persons [3]". The denotation d*ialogue* has to be distinguished from *conversation* and *discussion*, because a dialogue has specific characteristics: it is structured to contain a beginning, a flow and an end; it involves codified turn taking; it needs a temporal isotopy, it means that there is no time lag in the verbal interactions [4]. Spatial isotopy is not a dialogue condition: we can make a dialogue with a person at a distance, for example through a phone call. Speaking and writing are the two possible ways of a dialogue. As a result, instant messaging permits people to have a dialogue. However, a correspondence and an exchange of mails don't belong to the category 'dialogue' because there is a time lag in the exchanges. The dialogue is attributed to two speakers, Speaker1 and Speaker2, who obey the principle of turn taking as the role of the speaker and the interlocutor is successively attributed to Speaker1 and Speaker2 [5]. In the case of the man-machine dialogue, one of the two speakers is a machine and the natural language is substituted with data processing codes. This substitution involves bilateral exchanges into the natural language. However, it is a question of either a voice user interface or a user-friendly interface with speech. Systems of the former type understand the human language by responding to oral injunctions. Systems of the latter type react through pronouncing instructions. The man-machine dialogue has been under research in natural language processing and artificial intelligence for more than thirty years [6]. It is a central sector of activity in Language Industry. In fact, man-machine dialogue systems are distinguished by their mode of running which can be more or less by the machine [7].

When the restrictions on use are striking, the verbal interactions have a big knowledge gap, as the GUS system that is dedicated to online air ticket sales [8]. In such systems, the notion of script has a leading role because it determines the nature of interactions and their organization mode. Outgoing messages are those of the artificial speaker; they are preset according to the understanding of the user's intention. The incoming messages are those of the human speaker, yet it is very often proven that "every man-machine dialogue system has its limits, and, quickly, the user notices that the machine has understanding limits [3] ".

Communication systems in natural language of the first generation are fundamentally dedicated to database queries or to very specific tasks. They permit to formulate instructions in natural language and answer them in natural language. Interaction systems in natural language of the second generation are more developed. They can deal with linguistic phenomena that impede the system effectiveness. To detect the human speaker's intention, it is necessary, in particular, to deal with the inference, the reference and the meta-dialogue. The processing of the implicit meaning consists of highlighting the implicit consequences of a statement [9]. For example, the fact of saying *the door is open* does not produce the same reactions according to contexts. This can mean 'the door must be closed', for example because of a draught or on the contrary, 'the door must be open', because somebody is coming. The processing of the inference consists of implementing extra-linguistic knowledge and tools capable of detecting the implicit meaning of a speech in a man-machine dialogue system [10]. The question of reference is crucial in natural dialogue processing. First, there is the answer to the question of first and second personal pronouns that change the referent in turn taking and depend on the situation of utterance [11]. Then, there is the analysis of the references set which concerns the third personal pronoun, nominal groups and proper nouns [12]. They function differently; it depends on either their reference functions in an anaphoric or a deictic manner. In the first case, their reference is found in the left context (it is found in the right context if the function is cataphoric). In the second case, it is to be found in the utterance situation. The processing of the reference needs to implement the same resources as before if it doesn't require to use tools capable of identifying the referents as well [13]. The meta-dialogue concept refers to the phatic function of language. Its processing "consists of specific sub-dialogues

[2]There is no special use of the word 'dialogue', not the one that is used in the theatre repertoire.

of repeat requests (followed or not by particular instructions), of confirmation or validation requests in case of ambiguity, of processing of the interlocutor questions on the results of understanding or recognition, of put on hold and of the dialogue maintain and revival [7]". Man-machine dialogue systems of the third generation use works based on machine learning [14]. They permit, in very constrained operational framework, to perform more effectively certain tasks carried out in other system types [15].

There are many types of chatbots. We can subdivide them into two categories [16]: a) chatbots which make illusion but are proven ineffective in use; b) chatbots more performing, namely intelligent chatbots. Category a) chatbots can simply make two or three general exchanges, and then show their inability to process the information. Category b) chatbots are able to process the language information when the latter is applied to a limited sector. There are many implemented solutions to develop chatbots: 1) non-linguistic solutions; 2) weakly linguistic solutions; 3) quasi-linguistic solutions; 4) linguistic solutions; 5) strongly linguistic solutions.

Whatever the solutions, they can integrate technologies namely, on the one hand, speech-to-text, and, on the other hand, speech synthesis. In the first case, it is about converting automatically an oral message into a written one. In the second case, it is the contrary; a written message is automatically converted into an oral one. These two technological aspects are not presented here. The solution types 1 to 4, generally, correspond to a question-and-answer system, namely a question is asked by a human being and the answer is given by a machine. Type 5 solutions are considered to be more than a question-and-answer system, since it is about producing deeper verbal interactions between two human beings as one is a human being and the other is a machine.

Type 1 solutions are trivial because they are based on the preparation of ready sentences that correspond to requests which, once recognized, are associated with beforehand recorded answers. The first chatbots designed on this principle date from about fifty years ago [17]. Type 2 solutions are more developed since they use keywords that automatically identify in requests in order to associate them to concepts, which in their turn are associated with already made answers. A big number of chatbots are based on this principle. They can be reinforced by statistical tools that use the identified request-answer association [18]. Type 3 solutions are namely quasi-linguistic because they include linguistic analysis for the chatbot component which processes the understanding of the request, *cf. infra.* However, it is not the case for the formulation of the answer; it is an already made sentence. Type 4 solutions integrate the request analysis part of type 3 solutions and rely on the natural language generation, *cf. infra*. This technology permits to give a similar response in the content, but presents various shapes. Type 5 solutions have to allow the robot to simulate a conversation to the maximum by relying on verbal interaction scripts. It is to be conceived.

3 Linguistic Intelligence Contribution[3]

By definition, an intelligent chatbot automatically understands the incoming messages (a question) and it associates them automatically to outgoing messages (a response). It aims at simulating verbal interactions between two humans by replacing one of the speakers (typically, the one who gives answers). The chatbot performance depends on its ability to interpret the incoming messages and produce relevant informative outgoing messages. In addition, its performance is, also, conditioned by the speed of exchanging information between the user and the machine.

In order to improve the performance of a chatbot, it has to include the linguistic processing of incoming and outgoing messages. The processing of the incoming messages differs from the one of the outgoing messages since in the first case, it relies on natural language understanding of texts and in the second case, it relies on the natural language generation; that is to say two different technologies of natural language processing [7]. So far, natural language understanding and natural language generation technologies can share the same data model to process

[3]About the concept of linguistic intelligence and its relationship with the concept of artificial intelligence, *cf.* P.-A. Buvet. sous presse, "*Linguistique et intelligence*" in Linguistique et … Peter Lang

either the incoming or the outgoing messages. And even more, this has to involve a device associating an incoming message to an outgoing one.

In order to facilitate the flow exchange in the information processing set, there has to be a formal and common representation of the language data. Here, the propositional contents presentation of incoming and outgoing messages is used, in terms of predicate-argument structure to understand and generate messages in a homogeneous manner. The predicate-argument structure concept relies on Zellig S. Harris works [19], Maurice Gross [20], Pierre-André Buvet [21], Salah Mejri [22] and Robert Martin [23]. These works are inspired by the predicate logic suggested by Gottlob Frege [24].

In this theory, the concept of predicate is defined as a relationship in which the propositions, in its logical term, are analyzed as a link between entities, named arguments. This modeling differs from the Aristotelian modeling and gives place to the functional representation of the propositional content[4]: proposition => prédicat (argument$_i$).[5] [6]

Applied to language facts, the calculation of predicates leads to associate the contents of an utterance to predicate-argument structure. For example, the utterance *Le médecin soigne un patient* (*The doctor treats a patient)* is metalinguistically represented by the following predicate-argument structure: soigner (médecin,patient)[7]. Utterances are distinguished by the fact of having simple or complex propositional content. In the second case, nested predicates characterize the utterances. For example, in the utterance *Le médecin affirme au patient qu'il est guéri,* (*The doctor tells the patient that he is cured*) the propositional content integrates another propositional content. Hence, the utterance is represented metalinguistically by a predicate-argument structure mentioning a predicate nested in another predicate: affirmer(médecin,patient, guéri(patient)).[8]

A more sophisticated abstraction layer is also possible by replacing lemmatized forms of lexical units by their semantic class. Thus, both of the utterances above have the following metalinguistic representation:

SOIN(HUMAIN1,HUMAIN2)

AFFIRMATION(HUMAIN1,HUMAIN2,GUERISON (HUMAIN2)).[9]

To process the information of the incoming and outgoing messages, the choice of linguistic formalism is crucial. Thus, it is necessary to precise the type of the theoretical approach which underlies it. Theories about language are distinguished by their privileged subject of study. Therefore, to study the language facts, morphological theories have as an entry point the morphology, for example the theory named construction morphology [25]. It is the same thing for the syntactic theories, semantic theories, pragmatic theories and the theory of enunciation compared, respectively, to syntax, semantics, enunciation and pragmatics. Lexical theories assign a central place to lexicon in the linguistic analyses they produce. It is the case of those which rely on predicate-argument structures to represent language facts.

The effectiveness of a chatbot is based on the understanding of the incoming message and its adequacy with the outgoing message. From this point of view, the predicate-argument structures contribute to the association of an incoming message and an outgoing message in the chain of processing information by allowing the connection between the outcome of the interpretation of the incoming message and the beginning of the production of the outgoing message. The solution presented here is based on natural language understanding technology and a natural language generation technology, *cf. infra*. The junction point between these two technologies is the arrival

[4] The representation is namely functional because the predicate corresponds to an algebraic function, the argument is a variable.

[5] The letter i indicates the number of arguments in relation to a predicate, namely arity.

[6] Proposition => predicate (argument$_i$)

[7] To treat (doctor, patient)

[8] To tell (doctor, patient, cure (patient)).

[9] TREATMENT(HUMAN1,HUMAN2).
TELL(HUMAN1,HUMAN2, HEALING(HUMAN2)).

point of the first technology in the form of predicate-argument structure and the departure point of the second technology, also, in the form of predicate-argument structure. When the incoming message expresses a need, its metalinguistic representation is incomplete, namely one of its components is missing. The need expressed initially is satisfied in the outgoing message when the metalinguistic representation, which produces it, is saturated, namely the missing component of the other presentation is specified. The other elements of the two metalinguistic representations being similar are linked in a way that an association exists between the incoming message and the outgoing message. From this point of view, the predicate-argument structures are adapted, as metalinguistic representations, in order to make an association between the incoming message and the outgoing message.

The data model used to develop a chatbot grants an important role to the concepts of predicate and argument and to the relationship between them to automatically analyze language facts. The latter uses linguistic resources which correspond to local grammars, dictionaries and basic rules, *cf. infra*. The homogenization of the integration and the use of these resources in the system, which makes the conversational agent functions, creates the first constraint for the flow of information processing. Especially as, the processes of understanding and generating messages have opposite objectives. In the first case, the input is a text and the output is a metalinguistic representation. It is the opposite in the second case. The input is a metalinguistic representation and the output is a text. Moreover, it is necessary to associate the output of the natural language understanding to the input of the natural language generation. A human being conversation simulation represents a second constraint. From this point of view, the quality and relevance of the outgoing message is essential as well as the rapidity of the processing; the time of reply has to be about a second. Finally, the third constraint concerns the robustness of the system and its capacity to adapt to all kinds of computer science environment.

The chatbot relevance rate has been the subject of a quantitative and qualitative evaluation through the evaluation of two system components: on the one hand, the one which processes the understanding of the incoming messages, on the other hand, the one which is in charge of the outgoing messages. In both evaluations, we have to measure the noise, non-relevant information presented by the system as relevant, and silence, relevant information not selected by the system [26]. The first evaluation relies on the one used by labeling systems; it concerns the comparison of beforehand information identified by at least a person with the identified and qualified information by the system. It consists of manually labeling an incoming message corpus [7] and compares the manually labeling with the labeling of the same corpus by the semantic analysis engine. In that respect, noise and silence are measured in terms of precision and recall rates[10]. The results are as the following: Precision rate: 92.1 / Recall rate: 94.8.

The second evaluation consists of checking if the results given by the solutions are in accordance or not with the expressed needs[11]. The evaluation protocol requires a person who doesn't know the given results by the chatbot in order for the comparison to be independent. In addition, the evaluation consists of manually develop an outgoing messages corpus and compare it with incoming messages produced by the chatbot. It appears that 11% of the manually obtained data are missing, but only 4% of the automatically given data are not in the reference corpus. The failures of the automatic generation module are due to questions of synonymy which appears inadequately processed until now.

[10] It is useful to suggest the following formulations to calculate them: *Precision=true positive/true positive+false positive/Recall=true positive+false negative.* The terms 'true positive' and 'false negative' indicate respectively the non-relevant information but identified as relevant and the relevant information but not identified as it is.

[11] The second evaluation differs from the first one because the generation process does not match with the beforehand manually made tag process. The control over the data, that the natural generation method involves, confirms that the noise concept has no point in the evaluation.

4 Operating Mode of Conversational Agents

We present the architecture of the system which makes the chatbot functions.

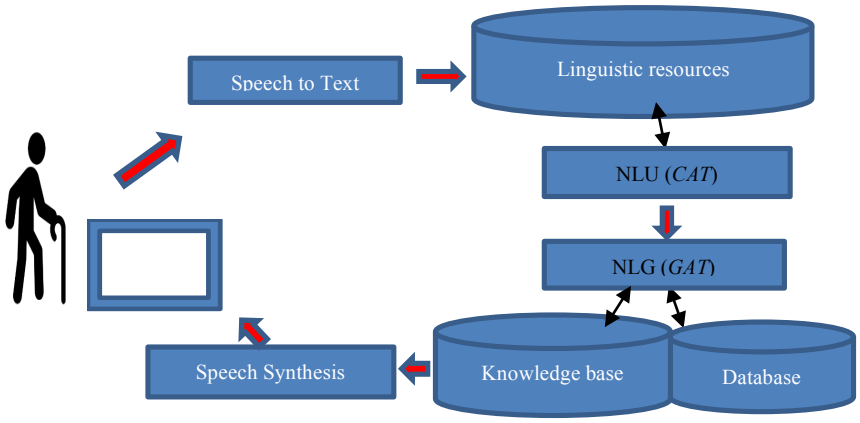


Fig. 1. The architecture of the system

API (Application Programming Interface) is the interface which permits the interactions between men and machines. Speech-to-text and speech synthesis modules have a role of transforming sound files into text files and another text file into a sound file. The Natural Language Understanding (NLU) module includes a semantic analysis engine which uses linguistic resources in the form of electronic dictionaries and local grammars [27]. The semantic engine's task is to replace the incoming message, in the form of text file, into a symbolic representation. This representation is given to the Natural Language Generation (NLG) module which associates the former representation to another symbolic representation from which an outgoing message will be produced in the form of text file.

The symbolic representation used in the system are metalinguistic representations, more precisely, it concerns semantic representations by making linguistic functioning mode of utterances. Incoming messages processed by the conversational agent have information requests or action requests. Information requests are introduced either in the form of questions (for example, *Que mangeons nous ce soir?* (*What do we eat this evening*?)) or in the form of requests (for example, *Je voudrais savoir ce que nous mangeons ce soir* (*I would like to know what we eat this evening.)).* Action requests are formulated in the form of orders (*Monte le chauffage (turn up the heating)*) or observations (*J'ai froid* (*I am cold)*). In the case of orders, the request is often formulated explicitly. In the case of observations, the request is implicit in a way that there has to be an inference in the observation about the need to be satisfied. Replies to action requests have two dimensions: extra-linguistic dimension, it concerns a concrete reply to the request in an appropriate way; and a linguistic dimension, it has to be known that the action requested is executed or about to be executed. Here, the extra-linguistic dimension is not developed.

When dealing with information or action requests, their metalinguistic representations include predicate-argument structures, *cf. supra*, in order to homogenize the process of information. Incoming messages are different in terms of the request type. In information requests, the human speaker's intention expresses an informative need which has to be satisfied. Also, the metalinguistic representation of an incoming message, henceforth RMi (*RMe*) has to specify this need in a way that can be associated with a metalinguistic representation of an outgoing message, henceforth RMo (*RMs*), relying on the fact that this representation can satisfy its need. Metalinguistic representations expressed, in terms of predicate-argument structures as functional representations, correspond to the following formula according to if the predicate is: a monadic predicate (*prédicat monadique*), namely it takes only

one argument, a unary operator (RM1 = PREDICAT (ARGUMENT1))[12]; a dyadic predicate (*prédicat dyadique*), namely it takes two arguments, binary operator (RM2 = PREDICAT (ARGUMENT1, ARGUMENT2))[13]; a triadic predicate (*prédicat triadique*), namely it takes three arguments, ternary operator (RM3 = PREDICAT (ARGUMENT1, ARGUMENT2, ARGUMENT3))[14]

The hypothesis has been made on the basis of the RMi which consists of merely one unknown element, a variable or a function, which is associated with the RMo from their other common elements which suggest the unknown element. Hence, the information request *Qu'est-ce qu'il a comme repas?* (*What do you have for lunch?)* has the following RMi MENU (X) and it is associated with the RMo MENU(%liste_plat%)[15] which is associated with the RMo MENU (%liste_plat%) because the predicate MENU appears in both representations in a way that the argument %liste_plat% is the object of the question represented by X. It is the same case with the information request *Est-ce que mon fils m'a appelé?* (*Did my son call me?)*, it has the following RMi OUI_NON=X (TELEPHONE (FILS (INTERLOCUTEURh) , INTERLOCUTEURh)[16] and it is associated, according to the context, to either the RMo OUI (TELEPHONE(FILS(INTERLOCUTEURh), INTERLOCUTEURh))[17]or to the RMo NON(TELEPHONE (FILS (INTERLOCUTEURh), INTERLOCUTEURh))[18] because (TELEPHONE (FILS (INTERLOCUTEURh), INTERLOCUTEURh))[19] appears in both of the representations in a way that the predicate OUI (yes) or NON (no) is the object of the question represented by the predicate OUI_NON=X .

In action requests, the human speaker's intention corresponds to a concrete need that has to be satisfied. An incoming message in a form of an order is sufficiently explicit to be returned to a RMi. However, when it is in a form of an observation, its intention is indirectly accessible to the interlocutor; there must be an inference of the semantic contents to express other semantic contents. The latter has to be added to the RMi in a form of a predicate-argument structure to the predicate-argument structure which has a link with the first semantic contents. There are inferences' rules used to result in such RMi. The association of these representations to the RMo is different from the previous one. It concerns the identification, among the second ones, of similar predicate-argument structures with the first ones and concatenates them with a predicate structure to execute the initially formulated request. For example, the action request *Il fait sombre?* (*Is it dark?)* has the following RMi FAIBLE LUMINOSITE (DEIXIS) & ORDRE (INTERLOCUTEURh, INTERLOCUTEURr, ALLUMAGE (LUMIERE)) and it is associated with the RMo FAIBLE LUMINOSITE(DEIXIS) & ORDRE (INTERLOCUTEURh, INTERLOCUTEURr, ALLUMAGE (INTERLOCUTEURr, LUMIERE)) & INFORMATION (INTERLOCUTEURr, INTERLOCUTEURh, EXECUTION ORDRE (INTERLOCUTEURh,INTERLOCUTEURr,ALLUMAGE (INTERLOCUTEURr , LUMIERE))).[20]The results RMo, for example to the following utterance is: *Comme il fait sombre, vous souhaitez que j'allume la lumière. C'est fait* (*Because it is dark, you want me to turn on the lights. It is done*).

The expressive language of semantic contents, an intention, comes from a huge diversity at the morphological, syntactical and lexical levels [28]. This transformation happens at the level of the text understanding module

[12](MR1 = PREDICATE (ARGUMENT1))
[13](MR2 = PREDICATE (ARGUMENT1, ARGUMENT2))
[14](MR3 = PREDICATE (ARGUMENT1, ARGUMENT2, ARGUMENT3))
[15](%list-dish%)
[16]YES-NO=X (TELEPHONE (SON (SPEAKERh), SPEAKERh)
[17]YES (TELEPHONE (SON (SPEAKERh), SPEAKERh))
[18]NO (TELEPHONE (SON (SPEAKERh), SPEAKERh))
[19](TELEPHONE (SON (SPEAKERh), SPEAKERh))
[20]LOW LIGHT (DEIXIS) & ORDER (INTERLOCUTORm, INTERLOCUTORr, TURN ON (LIGHT)) and it is associated with the RMo LOW LIGHT (DEIXIS) & ORDER (INTERLOCUTORh, INTERLOCUTORr, TURN ON (INTERLOCUTORr, LIGHT)) & INFORMATION (INTERLOCUTORr, INTERLOCUTORh, ORDER EXECUTION (INTERLOCUTORh, INTERLOCUTORr, TURN ON (INTERLOCUTORr, LIGHT))).

through two steps: 1) data extraction through the identification of their linguistic form; 2) qualification and interpretation of the data by associating them to semantic meta-information. To interpret these incoming messages, there has to be a semantic analysis engine which simulates three linguistic capacities made by humans. The first capacity is the lexical capacity, namely the memorization of simple or compound words. The second capacity is the structural capacity which concerns the morphological, syntactical and semantic levels of the language. The third capacity is the combinatorial capacity, namely to express the same propositional content in all kinds of ways. The simulation of these three capacities needs Information Technology tools and linguistic resources to identify the information and qualify it with semantic tags, *cf. infra*. The choice of adding meta-information to texts in terms of predicate-argument structure relies on the identified lexical units' properties. They have to be sufficiently described in the linguistic resources for the semantic labeling to operate.

The transformation of a RMo into an outgoing message is another difficulty. The choice has been made from a process which relies on conceptual dictionaries, namely dictionaries whose macrostructure is formed of concepts and the microstructure of lexical units associated with each concept. There are two conceptual dictionaries: the one of predicates and the one of arguments. The RMo are in form of predicate-argument structures as the predicate corresponds to the function and the arguments correspond to its variables. The predicates and their arguments are symbolized by their semantic class. Though the pragmatic adjustments, for example *Lundi* (*Monday*) when the incoming message is *Quel jour sommes-nous?* (*What day are we?*), the semantic class is related to the lexical units which they characterize. For example, the semantic class METEO_NEIGE[21] characterizes the verb *neiger* (*to snow)*, the noun *neige* (*snow)* and the adjective *neigeux (snowy).* The lexicon instances of a predicate, unlike the ones of arguments, have the particularity of being characterized by standard constructions. For example, the construction X0:PRONOM V[22] characterizes the verbal predicate *neiger (to snow) (il neige) (it snows),* the constructions X0:PRONOM y avoir DU N and X0:GROUPE_NOMINAL être à LE N[23] characterize the nominal predicate *neige* (*il y a de la neige* et *le temps est à la neige*) (*snow*) (*there is snow* and *weather forecast calls for snow*)) and the construction X0:PRONOM+GROUPE_NOMINAL être A[24] characterize the adjective predicate *neigeux (snowy)*. The unique argument, namely X0, is semantically specified at the distributional semantic level; it is the subject in all the constructions as deictic, nominal group or both of them. There are reconstructions associated with constructions, namely non-standard constructions or the component order is modified in a way that the syntactic modification does not involve the major semantic modification. For example, the construction X0:GROUPE_NOMINAL être à LE N[25] (*Le temps est à la neige*) (*weather forecast calls for snow)* has as a possible reconstruction, ce être à LE N que être X0:GROUPE_NOMNAL[26] (*C'est à la neige qu'est le temps*) (*it is called for snow by weather forecast*). The juxtaposition of these different metainformation gives rise to sentence patterns in the instantiation phase of the lexical units in the constructions, namely when the major components are replaced by a beforehand specified vocabulary. Morphosyntactic resources are, then, used to apply formal grammar to sentence patterns and produce well-formed utterances. Linguistic descriptions are formalized in electronic dictionaries and rule bases. Their good quality and systemicity are fundamental for the well-functioning of the Natural Language Generation module. The latter simulates the human language by suggesting all kinds of utterances relevant to the same semantic contents. The objective is met because the lexical and syntactic variety is taken into consideration by the data model implemented by the system.

[21] WHEATHER FORECAST_SNOW
[22] X0: PRONOUN V
[23] X0: PRONOUN have of N and X0: NOMINAL_GROUP to be at THE N
[24] X0: ARTICLE+NOMINAL_GROUP to be A
[25] X0: NOMINAL_GROUP to be at THE N
[26] This to be at THE N that to be X0: NOMINAL_GROUP

In the Natural Language Understanding module, the incoming messages are enhanced with semantic tags, dictionaries and *local grammars*. Below, we introduce an excerpt of one of these dictionaries, the body-parts dictionary. This dictionary contributes to the identification and qualification of information relevant to the health of the conversational agent users.

```
annulaire,.N+H_PARTIE_CORPS
(ring finger,. N+H_BODY_PART)
annulaires,annulaire.N+H_PARTIE_CORPS
(ring fingers,ring finger. N+H_BODY_PART)
articulation,.N+H_PARTIE_CORPS
(joint,. N+H_BODY_PART)
articulations,articulation.N+H_PARTIE_CORPS
(joints,joint. N+H_BODY_PART)
avant bras,avant-bras.N+H_PARTIE_CORPS
(forearms,forearm. N+H_BODY_PART)
avant-bras,.N+H_PARTIE_CORPS
(forearm,. N+H_BODY_PART)
auriculaire,.N+H_PARTIE_CORPS
(auricular,. N+H_BODY_PART)
auriculaires,auriculaire .N+H_PARTIE_CORPS
(auricular,auricular. N+H_BODY_PART)
bouches,.N+H_PARTIE_CORPS    (mouths,. N+H_BODY_PART)
```

Fig. 2. A dictionary relevant to the health of the conversational agent users

The information on the left side of the full stop is of a linguistic nature and the one on the right side of the full stop is of a meta-linguistic nature. The macrostructure of the dictionary is made of lexical units constituting the dictionary. The microstructure is constituted of a lexical unit followed by a comma and its lemmatized form if the entry is an alternative. Next, there is a full stop which is mandatory followed by a grammatical category (N is the code for noun) and by meta-information of a semantic nature, in this case, the code H_PARTIE_CORPS (H_BODY_PART) is relative to body part hyperclass. This meta-information may be called from a local grammar in a way that all the dictionary items are recognized.

Below, we introduce one of these local grammars, the local grammar namely PROBLEME_SANTE (HEALTH_PROBLEM).

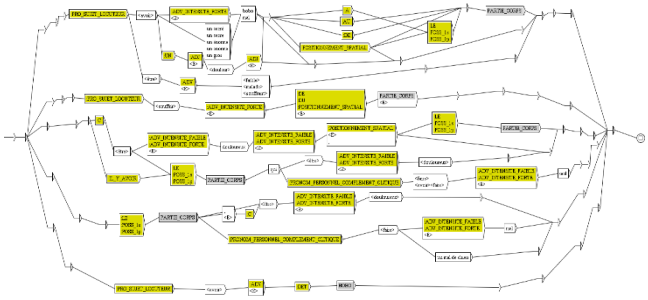


Fig. 3. Local grammar (HEALTH_PROBLEM)

Local grammars are formal representations of contextual elements. Regular grammars used by a parser which constitute the simplest grammar class in Chomsky hierarchy [29]. The representation of a finite state automaton is the one of an oriented graph (namely with a departure point and an arrival point) in which the windows are the states and the arrows are the transitions between these states. They correspond to formalized descriptions of syntax of a semantic class or a grammatical class. They are implemented in forms of automata which run through texts in order to identify and qualify the information. Local grammars use electronic dictionaries which are formal representations of lexicographical data. They are introduced in forms of graphs involving a start node and an end node and information nodes of lexical or morphological nature, the nodes linked make different possible combinations, regressions and repetitive structures. A graph can call other graphs which makes its combinatorics even more powerful. In computer science, graphs correspond to either finite state automata or finite state transducers. Finite state automata allow the identification of information by running linearly through texts and by reporting each connection between the text and one of its graphic paths. Finite state transducers integrate the first ones; they identify information according to the same principles and qualify them by inserting new data.

PROBLEME_SANTE (HEALTH_PROBLEM) graph calls other graphs and one of them is the PARTIE_CORPS (BODY_PART) graph which represents the dictionary PARTIE_CORPS. It allows the identification of all kinds of utterances which have a relationship with a health problem related to the human speaker, for example, *J'ai mal à la poitrine* or *Mon ventre me fait mal* (*I have pain in my chest* or *My stomach hurts).* It qualifies them, by adding a tag, in a predicate-argument structure form PROBLEME_SANTE (LOCUTEUR_HUMAIN)[27].

The information specified in the utterance is not always explicit. For example, in the use of a chatbot context, reporting a physical pain implies the implicit information '*j'ai besoin d'être soigné*' ('I need to be treated'). It follows that the tag PROBLEME_SANTE (LOCUTEUR_HUMAIN) is interpreted as an action request and it has to be completed with the tag APPEL (LOCUTEUR_MACHINE,PERSONNEL_SOIGNANT)[28]. Such transformations happen when the RMo are given to the Natural Language Generation module and they are processed by inference rules.

In the Natural Language Generation module, the semantic representation of the outgoing message is a predicate-argument structure, *cf. supra*. From this representation, it is possible to generate a well formulated utterance by applying well ranked rules which consider: semantic properties of a distributional and lexical nature; syntactic properties related to the predicate constructions and the positioning of the lexical material, specified in a previous rule, in these constructions (a procedure named *instanciation* (instantiation)); enunciative properties relative to the actualization of predicates and arguments (a procedure named *actualisation* (actualization)); morphosyntactic

[27] HEALTH_PROBLEM (HUMAN_SPEAKER).
[28] CALL (MACHINE_SPEAKER, NURSING_STAFF).

properties which give rise to verb conjugation and coordinating conjunction rules with adjectives and nouns; pragmatic properties which narrow the field of possibilities and take the extra-linguistic context into consideration.

The semantic, syntactic and enunciative properties are listed in a conceptual dictionary. This dictionary is transferred to the database. The fourth category of properties is about the morphosyntactic dictionary, created as part of the project. This second dictionary is also inserted in the database. The fifth category of properties is covered by the knowledge base.

These linguistic and extra-linguistic resources are used by the natural generation module through the implementation of a set of rules which use linguistic resources inserted in the database. Below, we list the different rules used. The order of their presentation corresponds to their order in the passages before.

Predicate identification rule

Semantic properties of a distributional nature define propositional contents as the predicates and their arguments which are specified in a functional representation form as a semantic value. For example, utterances such as *Il y a du brouillard, C'est brouillardeux* and *Le temps est au brouillard* (*There is fog,* It is *foggy* and *The weather forecast calls for fog)* share the same semantic contents represented as the following: METEO_BROUILLARD (DEIXIS+METEO)[29]. The first rule used by the natural generation module relies on this type of meta-linguistic description.

The departure point of the generation module is a predicate-argument structure. Its functional representation indicates if we are dealing with a simple predication or a complex predication. The second predication is distinguished from the first one by the fact that it incorporates at least another simple or complex predication, in its argument domain. The functional representation of a complex predication involves at least a double parenthesis, which allows its automatic identification. For example, the utterance *Le médecin est absent* (*The doctor is absent)* is analyzed as a simple predication which corresponds to the following predicate-argument structure: 1) (ABSENCE (HUMAIN:MEDECIN))[30].

Furthermore, an utterance as: *Le médecin a déclaré aux patients qu'il était absent* (*The doctor announced to his patients that he was absent)* is analyzed as a complex predication which corresponds to the following predicate-argument structure: 2) (DECLARATION(HUMAIN:MEDECIN,HUMAIN:PATIENT,ABSENCE(HUMAIN:MEDECIN)))[31].

The end of the representation 1) involves one closing parenthesis while the representation 2) involves two closing parentheses. The number of the closing parentheses indicates the type of the predication. If it is simple, it includes one closing parenthesis. If it is complex, it includes more than one closing parenthesis.

First, it is a question of the rules related to the simple predication.

Distributional rule

This rule specifies the information which considers the predicate-argument structure by explicitly specifying the semantic nature of predicates and the different categories of arguments. There are three categories of predicates, monadic predicates, dyadic predicates and the triadic predicates, *cf. supra*[32]. It follows that there are three categories of arguments: the first argument which is coded X0 and it is observed with the three categories of

[29] WEATHER REPORT_FOG (DEIXIS+WEATHER REPORT).

[30] ABSENCE (HUMAN:DOCTOR).

[31] ANNOUNCEMENT(HUMAN:DOCTOR,HUMAN:PATIENT,ABSENCE(HUMAN:DOCTOR)))

[32] There is a fourth category of predicates, a predicate that its domain of arguments is quaternary. Its scarcity led to its neglecting [30].

predicates, the second argument, coded X1 and which is observed with the dyadic and triadic predicates, the third argument, coded X2 which is observed merely with one triadic predicate. The three predicate categories are respectively coded PRED(X0) monadic predicate, PRED(X0,X1) dyadic predicate and PRED (X0,X1,X2) triadic predicate. For example, the utterance *La météo est bonne* (*The weather is fine)* is generated from the following information: PRED=METEO%type_temps%(X0=METEO+X0=DEICTIQUE)[33].

The code %type_temps% means that the information is not in the linguistic database but depends on the knowledge base. For the X0 argument, it is about either a substantive of the METEO (WEATHER FORECAST) class or about a pronoun of the DEICTIQUE (DEICTIC) class. The exact nature of the deictic is specified at the level of the constructional rule (*règle constructionnelle*), *cf. infra*.

Pragmatic Adjustment rule

To limit the number of answers, the knowledge base is requested straightaway to identify a precised weather forecast type. As a result, the previous rule is rewritten as the following (in the case of a fine weather): PRED=BEAU_TEMPS(X0=METEO+X0=DEICTIQUE)[34].

Lexical rule

It is about the specification of the linguistic units associated with the semantic categories of predicates and arguments. For example, for the predicate class BEAU_TEMPS (FINE_WEATHER), the rule produces the following linguistic forms: the nouns *beau temp*s, *temps magnifique* (*fine weather, a nice day);* the adjectives *agréable*, *beau, bon*, *magnifique* (*pleasant, delightful, nice);* or a verbal phrase *aller vers le beau* (*it is going to be beautiful)* and for the argumental class METEO (WEATHER FORECAST) the nouns *météo* and *temps* (*weather forecast* and *weather).*

Constructional rule

It is about the specification of the constructions connected to the linguistic forms of predicates. For example, in *beau temps* (*fine weather)*, it is about the construction X0 :DEIXIS1 *avoir* DU N[35] as N is for *beau temps* (*fine weather)*[36], and the subject is the argument X0 of the nominal predicate under the value coded DEIXIS1, which means *on* or *nous* (*we).*

Reconstruction rule

The construction, associated with the linguistic form of a predicate, is canonical; it involves a standard layout of the subject position and, if any, the positions of the first and second complement occupied by the arguments. In speech, this layout may be changed and codified [31]. It is the case with the passive form in comparison to the active form of a sentence. For example, the predicate *petit déjeuner* (*breakfast)* is associated with the canonical construction: LE N *être* X0:GN[37] (*Le petit déjeuner est du café, des tartines et un jus d'orange*) *(The breakfast is*

[33] PRED=WEATHERFORECAST%type_weather%(X0=WEATHER FORECAST+X0=DEICTIC)
[34] PRED=FINE_WEATHER (X0=WEATHER FORECAST+X0=DEICTIC)
[35] X0: DEIXIS1 to have THE N
[36] *Avoir* (*to have*) is a helping verb of the nominal predicate [20].
[37] THE N to be X0:NG

coffee, a toast and an orange juice) and with the reconstructions *il y avoir* comme N X0:GN[38] (*Il y a comme petit déjeuner du café, des tartines et un jus d'orange*) *(There is for breakfast: coffee, a toast and an orange juice)*, etc.

The lexical variety of the predicates and arguments and the constructional variety, combined with each other, explain that from the same propositional content, the system produces a huge number of utterances and allows the stimulation of a human conversation, *cf. infra*.

Instantiation rule

This rule consists of merging the three previous rules and inserting the linguistic forms in the positions of the predicates and arguments. For example, in the previous example, the construction *LE N être X0: GN* is transformed into LE *petit déjeuner être* DET *café,* DET *tartines,* DET *jus d'orange*[39], the reconstruction *il y avoir comme* N X0:GN into *il y avoir comme petit déjeuner* DET *café,* DET *tartines,* DET *jus d'orange* [40], etc.

Actualization rule

It is about the specification of the predicate time and the argument determination. For example, when the instantiation rule gives such results : *aujourd'hui,* DET *météo être à la pluie/aujourd'hui,* DET *temps être à la pluie/*DET *météo être à la pluie aujourd'hui/*DET *temps être à la pluie aujourd'hui*[41]. The application of the code relative to the enunciative properties of the nominal predicate *pluie* (*rain*) produces the result : *aujourd'hui,* LE *météo être:*PRESENT *à la pluie/aujourd'hui,* LE *temps être:*PRESENT *à la pluie/*LE *météo être:*PRESENT *à la pluie aujourd'hui,* LE *temps être:*PRESENT *à la pluie aujourd'hui*[42].

Conjugation and coordinating conjunctions rule

This rule specifies how to conjugate a verb, how to coordinate and use determiners.

Morphological adjustment rule

This rule specifies in which conditions, there are contractions, transformations or removal of concatenated forms and stop words (*mots vides*), for example:

Lexico-syntactic adjustment rule

The role of this rule is to limit the descriptive power of the syntactic properties, particularly the ones which concern the reconstructions and/or the lexical combinatorics. In fact, all the predicate uses defined by the same construction do not accept the reconstructions associated with them. Hence, it is convenient to limit them by only specifying the ones applicable. For example, adjective predicates *bon*, *mauvais*, *pourri (good, bad, rotten)* are characterized by the construction X0: GN *être* A (X0: NG *to be* A). This allows the reconstruction X0: GN *être*

[38] There to be as N X0:NG

[39] THE breakfast to be DET coffee, DET toast, DET orange juice

[40] There to have as breakfast DET coffee, DET toast, DET orange juice

[41] Today, DET weather forecast to be at the rain/today, DET weather to be at the rain/DET weather forecast to be at the rain today/DET weather to be at the rain today

[42] Today, THE weather forecast to be: PRESENT at the rain/today, THE weather to be: PRESENT at the rain/ THE weather forecast to be: PRESENT at the rain today, THE weather to be: PRESENT at the rain today.

très A and X0:GN *être vraiment* A (X0: NG *to be very* A and X0 : NG *to be really* A) in a way that the following utterances can be generated: *La météo est bonne*, *La météo est très bonne*, *La météo est vraiment bonne*, *Le temps est bon, Le temps est très bon*, *Le temps est vraiment bon*, *La météo est mauvaise*, *La météo est très mauvaise*, *La météo est vraiment mauvaise*, *Le temps est mauvais, Le temps est très mauvais*, *Le temps est vraiment mauvais*, *La météo est pourrie*, *La météo est très pourrie*, *La météo est vraiment pourrie*, *Le temps est pourri, Le temps est très pourri*, *Le temps est vraiment pourri*. (*The weather forecast is good, The weather forecast is very good, The weather forecast is really good, the weather is good, the weather is very good, the weather is really good, the weather forecast is bad, the weather forecast is very bad, the weather forecast is really bad, the weather forecast is rotten, the weather forecast is very rotten, the weather forecast is really rotten).* Yet, the utterances *Le temps est bon, Le temps est très bon*, *Le temps est vraiment bon (the weather is good, the weather is very good, the weather is really good)* are not acceptable (lexical combinatorics constraint) and *La météo est très pourrie (the weather forecast is rotten)* (reconstruction constraint). In addition, it is convenient to precise these constraints in a lexico-syntactic adjustment rule.

In the case of a complex predication, the same rules are applied. However, once the rule of the predicate identification is applied, they have to process the *embedded predicate before the predicate which embeds a unit. When there are more than two predicates which are embedded, those rules must be applied from the most embedded predicate to the one whose level of embedding is the highest, cf. supra.*

5 Conclusion

The works that have been carried out as part of the chatbot development have contributed to set the foundations of a lexical generative grammar of the French language. The development of this grammar will allow the creation of other chatbot processing others themes.

References

1. Asimov, I.: Le cycle des robots. J'ai lu, Paris (2001).
2. Devillers, L.: Des robots et des hommes: mythes, fantasmes et réalité. Plon, Paris (2017).
3. Landragin, F.: Dialogue homme-machine conception et enjeux. Hermès science publications-Lavoisier, Paris (2013).
4. Kerbrat-Orecchioni, C.: Les interactions verbales. Armand Colin, Paris (1990).
5. Sacks, H., Schegloff, E.A., Jefferson, G.: A Simplest Systematics for the Organization of Turn-Taking for Conversation. Language. 50, 696–735 (1974).
6. Sabah, G.: L'intelligence artificielle et le langage. Hermès, Paris (1989).
7. Pierrel, J.-M. ed: Ingénierie des langues. Hermès Science publications, Paris (2000).
8. Vilnat, A.: Quels processus pour les dialogues homme-machine. In: Sabah, G. (ed.) Machine, langage et dialogue. L'Harmattan, Paris (1997).
9. Henze, N., Dolog, P., Nejdl, W.: Reasoning and ontologies for personalized e-learning in the semantic web. Educational Technology & Society. 7, 82–97 (2004).
10. Hoc, J.-M.: Psychologie cognitive de la planification. Presses universitaires de Grenoble, Grenoble (1987).
11. Benveniste, É.: Problèmes de linguistique générale. (1966).
12. Corblin, F.: Les formes de reprise dans le discours: anaphores et chaînes de référence. Presses Univ. de Rennes, Rennes (1995).
13. Grosz, B., Sidner, C.: Attention, intentions, and the structure of discourse, (1986).
14. Hiraoka, T., Neubig, G., Yoshino, K., Toda, T., Nakamura, S.: Active Learning for Example-Based Dialog Systems. In: Jokinen, K. and Wilcock, G. (eds.) Dialogues with social robots. pp. 67–78 (2017).

15. Chandramohan, S., Geist, M., Lefèvre, F., Pietquin, O.: User Simulation in Dialogue Systems Using Inverse Reinforcement Learning. (2011).
16. Gouritin,T.:L'arnaque chatbots durera-t-elle encore longtemps?, https://www.frenchweb.fr/larnaque-chatbots-durera-t-elle-encore-longtemps/305697, (2018).
17. Weizenbaum, J.: ELIZA - Un programme informatique pour l'étude de la communication en langage naturel entre l'homme et la machine. Communications de l'ACM (1966).
18. Bisk, Y., Yuret, D., Marcu, D.: Natural Language Communication with Robots. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 751–761. Association for Computational Linguistics, San Diego, California (2016).
19. Harris, Z.S.: Notes du cours de syntaxe. Seuil (1976).
20. Gross, M.: Les bases empiriques de la notion de prédicat sémantique. Langages. 15, 7–52 (1981).
21. Buvet, P.-A.: La dimension lexicale de la détermination en français. Honoré Champion, Paris (2013).
22. Mejri, S.: Le prédicat et les trois fonctions primaires. In: Souza Silva Costa, D. and Bençal, D.R. (eds.) Nos caminhos do léxico. Editora UFMS, Campo Grande do Sul (2016).
23. Martin, R.: Linguistique de l'universel: réflexions sur les universaux du langage, les concepts universels, la notion de langue universelle. Académie des inscriptions (2016).
24. Frege, G., Imbert, C.: Écrits logiques et philosophiques. Seuil, Paris (1971).
25. Fradin, B.: Nouvelles approches en morphologie. Presses Universitaires de France, Paris cedex 14 (2003).
26. Shannon, C.E.: A Mathematical Theory of Communication. Bell System Technical Journal. 27, 379–423 (1948).
27. Buvet, P.-A.: Compréhension automatique des articles politiques : le traitement des discours rapportés. Ela. Études de linguistique appliquée. 180, 475–492 (2015).
28. Mel′čuk, I.: Cours de morphologie générale: théorique et descriptive. Presses de l'Université de Montréal, Montréal (1993).
29. Chomsky, N., Miller, G.A.: L'analyse formelle des langues naturelles. Mouton, Paris (1968).
30. Leclère, C.: Travaux récents en lexique-grammaire des verbes français. Duculot (1998).
31. Muller, C.: Le prédicat, entre (méta)catégorie et fonction. (2013).

Hybrid Question Answering System based on Natural Language Processing and SPARQL Query

Mickael Rajosoa, Rim Hantach, Sarra Ben Abbes, Philippe Calvez
CSAI LAB ENGIE, France
Rajosoa.Mickael@gmail.com, Rim.Hantach@external.engie.com,
Sarra.BEN-ABBES@external.engie.com, Philippe.Calvez1@engie.com

Abstract

Chatbot is a conversational agent that communicates with users based on natural language. It is founded on a question answering system which tries to understand the intent of the user. Several chatbot methods deal with a model based template of question answering. However, these approaches are not able to cope with various questions and can affect the quality of the results. To address this issue, we propose a new semantic question answering approach combining Natural Language Processing (NLP) methods and Semantic Web techniques to analyze user's question and transform it into SPARQL query. An ontology has been developed to represent the domain knowledge of the chatbot. Experimentations show that our approach outperforms state of the art methods.

1 INTRODUCTION

Nowadays, the huge amount of data has been increased which makes the task of information retrieval more difficult. To overcome this problem, several approaches in different domain areas have been proposed based on question answering system (QA) [BAB12, YD14a] where the aim is to understand natural language questions and extract relevant information. QA is a computer science discipline designed to generate an answer of question posed by human in natural language. This system is based on Natural Language Processing methods and Information Retrieval (IR) [CLR13, Fer16]. NLP helps computer to understand and answer user's query through IR. Furthermore, QA systems are divided into two categories: an open domain and a closed domain. The open domain deals with questions related to several topics and the closed domain deals with questions related to a specific domain. In order to represent knowledge and facilitate the information retrieval, Semantic Web techniques are required. Ontology is one of these techniques. It's a formal model that allows to describe concepts and relations between them in a domain. However, recent works in the literature have not highly developed these techniques, they rely on keywords to identify the context and answers for user's question. These methods involve the removal of stop words. Nevertheless, the removal of these words can lead to the loss of the sentence's meaning. From this point, we found some improvement areas which motivate us to combine NLP methods and Semantic Web techniques. The main objective of our system is to get user's intent based on syntactic dependency relationships. In this paper, we review the related work in section 2. Then, in section 3, we highlight the use of the syntactic relationships to translate user's question into triple patterns and build the SPARQL queries. We conduct a comparative study with previous related work to prove the effectiveness of our approach in section 5. Finally, in section 6, we provide some conclusions.

Copyright © by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2 Related Work

Several chatbot approaches have been addressed in the literature where most of them are based mainly on the preparation of a question answer template. In [A+18], authors suggest a chatbot leaded and controlled by template questions. Therefore, when user asks a question and it is present in the file that contains all the templates for questions and answers (AIML), the bot can provide an answer based on the question template. However, this approach, as shown in many works, has revealed a number of unexpected problems and weaknesses related to the reliability of answers. In practise, it is impossible to list all the possible questions that a user may have. Therefore, if a user's question is not in the dataset, the bot can't supply an answer.

Researchers proposed tools to transform user's question into SPARQL query language in order to find the answer. They created an application called Quepy [Mac18][BC14]. The purpose of this application is to transform a question (in natural language) into SPARQL in order to query linked open data such as DBpedia or Freebase [ABK+07, YD14b]. In this application, they highlight the use of NLP methods to identify named entities (i.e. human entities, places, organizations) and question templates in regular expression form to generate the SPARQL query. However, such an approach is doomed to be ineffective because it is based on a prepared template. To resolve these limitations, researchers advanced more in-depth approaches. For example, C. S. Kulkarni et al. [KBPK17] propose a new NLP and machine learning approach to cope with agent conversational system problems. First, a dataset of questions/answers has been prepared in order to train the model. Then, to categorize users question, authors suggested a non supervised classification algorithm where a cosine similarity measure has been used to classify new users questions. However, this approach, while quite efficient, does not deal with semantic relationships between questions.

In [BDNM18], a new approach has been proposed based on the combination of Semantic and NLP methods to enhance chatbots ability. Giving a question, keywords and named entities have been extracted. Therefore, the keywords express the intent of the question and help in the construction of the SPARQL request. In addition, authors propose to use WordNet [Mil95] to establish a synonym list and perform mapping. Nevertheless, removing the stop words and extracting only the keywords in order to identify questions intent can severely limit its effectiveness and induce erroneous answers. In [AKS17], A. Albarghothi et al. combine as well a linguistic approach with Semantic Web processing. They use NLP functions (normalization, tokenization, removing stop words, stemming, tagging) to translate natural language into triple patterns and query an ontology. This approach is limited because it suffers from semantic rules. In [APMG12], authors establish an ontology and AIML categories to reply users question. After a classic processing of the users question, they convert properties and relations between concepts into AIML categories to supply a complete sentence for the user. Their approach requires improvements to be deemed in industry. In [SWRR14], a simple Knowledge Organization System (SKOS) and Spin rules have been used to translate natural language into SPARQL. Nevertheless, this method is not yet applicable under industrial conditions.

In [NNBU+13], a new approach has been established to transform a SPARQL request into a natural language. To do this, different rules (these rules look like patterns) have been defined to build the sentences. Unfortunately, this approach suffers from the syntactic and semantic aspects. In fact, generated sentences may poorly be tuned due to the superimposition of rules. Thus, we obtain as results, sentences having no sense or without correct grammar rules..

Different state of the art approaches suffer from semantic and syntactic relationships to understand the intent of the question. To overcome these limitations, we propose a new semantic chatbot based on the dependency relationships and the SPARQL query. The originality of our approach lies in the definition of new rules to deal with question answering system.

3 Proposed approach based on Linguistics and Semantics rules

The approach is based on the combination of NLP methods and Semantic Web techniques. The purpose of this combination is to understand user's question. Here, "understand" means to get user's intent (what is he looking for behind his question?) in order to supply a correct answer. Our approach requires an ontology to represent information and relationships related to the topics. The main step, is to analyze the words of a sentence. It should be emphasized that words part of speech is not sufficient to understand the meaning of a sentence. It is also necessary to deal with the syntactic function of different words and detect the named entities which can help us to perceive the meaning of the question and discern the intention. Thus, we use the NLP tools to extract the syntactic structure of the sentence. Then, this linguistic information will be used to build the rules. Finally, these rules will be transformed into SPARQL queries to request the triple store. In addition to that, external

resources such as WordNet and DBpedia have been used to enhance and strengthen the reliability of our chatbot. Figure 1 shows an illustration of the proposed approach.

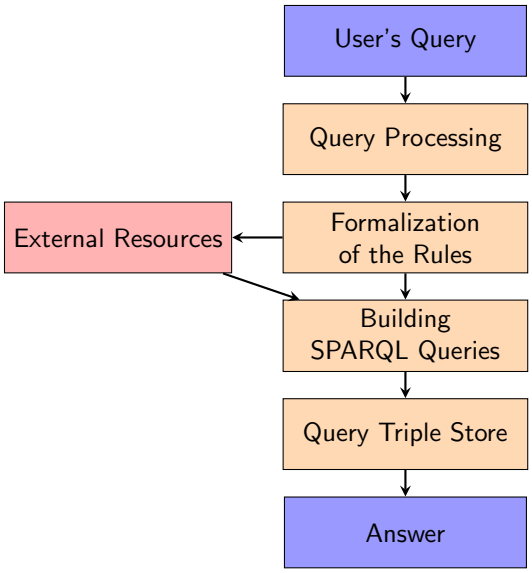


Figure 1: Process of the proposed approach

We start by the query processing (syntactic and linguistic annotations of the question), then, rules' formalization in order to transform the question into SPARQL query and finally, we query the triple store to get the answer.

3.1 Query Processing

The aim of the query processing is to analyze the linguistic and syntactic structure of the question. By focusing on the structure, we can know: what is the subject of the question? What is the core of the question or the main verb? What are the complements (object, noun)? Does the sentence have any specific feature for example coordinating conjunctions, a question with a copula (an intransitivity verb which links a subject to a noun phrase, adjective or other constituent which expresses the predicate)? Moreover, we start by extracting the part of speech of each word which allows us to know the subject of the sentence, the main verb that symbolizes the action of the sentence and the complements for specific cases. Then, we extract the named entities to identify the context of the sentence. Finally, the dependency relationships have been identified in order to obtain the syntactic functions and get relations between words in a sentence.

3.2 Formalization of the rules

During this step, we develop a generalized rules model based on syntactic relationships in a question. The defined rules could be adapted to any question (who, what, where, when, how, etc.), they are presented as follows:

- 1st rule: when the question contains a possessive phrase. This means that the possessive will give additional information to his syntactic head. Thus, this syntactic head will become the secondary predicate.

  Example : What is the name of Michelle Obama's daughters ?

  "name" represents the main predicate and the apostrophe "s" is the possessive of "Michelle Obama". The syntactic head of this possessive is "daughters". Therefore, the latter become the secondary predicate.

- 2nd rule: when the question contains coordinating conjunctions such as "and", "or". This means that all named entities in the question are on the same level. In other words, they share the same main predicate.
  Example: Who are the daughters of Michelle Obama and Barack Obama ?
  "daughters" is the main predicate and we have two named entities which are separated by a coordinating conjunction "and" which means that we want to know the daughter of both.

- 3rd rule: when the question contains a non-verbal predicate, for example the verb "to be" and it is preceded by the main predicate and it's an interrogative adverb type, then, the intent of the question will be a nominal subject (in most cases it's a noun).

  Example : How is Eagle's syndrome ?

  In this example, we have an interrogative adverb "How" followed by a non-verbal predicate. Consequently, we focus on the nominal subject of the question to get the intent. "syndrome" is the subject of the question. So, it becomes the main predicate.

- 4th rule: other questions that have a subject, a main predicate and a complement.

  Example : Who wrote Harry Potter ?

  "wrote" is the main predicate of the question and "Harry Potter" is a named entity and a complement. In this case, we are looking for the subject of the question. In other words, the writer of the novel.

These rules can be perfectly combined. If a sentence or a question follows a rule, it does not exclude the other rules. The main issue of syntactic dependencies is the identification of user's intent. Indeed, sometimes NLP methods are not able to correctly identify the intent. That's why, external resources have been used to deal with semantic relations (section 3.5).

3.3 Building SPARQL Queries

After listing all the rules (section 3.2), we now associate each of these rules to a SPARQL request. The core of the question will be considered as the main predicate. In other words, it's the main property that will connect a resource A to a resource B, in each question there is always a main predicate. For the modifiers or complements, their syntactic heads will be considered as the secondary property. This property will link a resource C to a resource A.

For the specific cases: first, we know that the named entities in a user's question share the same property when we have a coordinating conjunction in the sentence. In other words, user's intent is exactly the same for these entities. Therefore, we use *UNION* structure because it's useful for concatenating solutions from two possibilities. Second, if the heart of the question is an interrogative adverb then we consider that the main predicate will be the nominal subject of this sentence. SPARQL queries for the different rules are defined as follows:

- 1st rule:

```
SELECT DISTINCT ?a
WHERE
    {?ans onto:main_predicate ?a
     ?x onto:secondary_predicate ?ans}
```

- 2nd rule:

```
SELECT DISTINCT ?ans_label
WHERE
    {?x onto:main_predicate ?ans
     ?y onto:main_predicate ?ans
     ?ans rdfs:label ?ans_label.
    {?x rdf:type onto:Person
     ?x rdfs:label 'X'.}
UNION
    {?y rdf:type onto:Person
     ?y rdfs:label 'Y'.}}
```

- 3rd rule:

```
SELECT DISTINCT ?ans_label
WHERE
    {?x onto:nominal_subject ?ans
     ?ans rdfs:label ?ans_label.}
```

- 4th rule:

```
SELECT DISTINCT ?a
WHERE
    {?ans onto:main_predicate ?a}
```

3.4 Query the Triple Store

The knowledge graph is stored in a triple store called GraphDB [NAJ14]. We chose a triple store because the data will be structured as a triplet and it will be easy to update it using SPARQL. We use a wrapper service to query the repository of our knowledge Graph and get answers to our questions (Figure 2).

User: Who is Camille Dupond's father?
Bot: Paul Dupond.
User: Where did Chantal come from?
Bot: Berlin.

Figure 2: Conversation between the bot and the user

3.5 External Resources

We employ external resources to reduce ambiguities. In fact, the user may use specific terms in his question and NLP tools are not able to identify the user's intent or extract the named entities. These problems are solved through two external resources: WordNet and DBpedia.

3.5.1 WordNet

WordNet [Mil95] is a lexical database developed by Princeton University. Once the intention of the question has been identified, a list of synonyms must be drawn up to promote mapping on the ontology. The integration of this resource allows our system to avoid ambiguities.

3.5.2 DBpedia

DBpedia is used to find the type of named entities. Indeed, NLP tools may omit to extract some entities. Thus, thanks to grammatical analysis which gave us upstream the different proper nouns of the sentence. DBpedia is able to identify the type of each proper name. It's a knowledge base that standardizes the content of Wikipedia. Each Wikipedia page is browsed by a set of extractors and these extractors will identify elements of the page and generate data. We map the proper noun with his label, then we try to get his type with a SPARQL request.

4 ILLUSTRATIVE WITH EXAMPLE

For our approach, we use Stanford Core NLP [Cor19] as NLP tools. In Table 1, we mention the main syntactic relations that interest us. As we can see on the left of the table, the annotation used by Stanford and on the right, names of syntactic functions in dependency grammar. We illustrate our approach using the example below.

Example: What is Genghis Khan's real name?

4.1 Query Processing

During this step, user's query follows four processing : tokenization, parsing, dependency parsing and named entity recognition (NER). Tokenization is the task of cutting it up into pieces, called tokens. Parsing gives the parts of speech of each word and the structure syntagmatics of sentence. Dependency Parsing analyzes the grammatical structure of a sentence, and establishes relationships between "head" words and words which modify those heads. NER classifies named entities that are present in a question into predefined categories like *person*, *organization*, *location*, etc..

Table 1: Main dependency relationships for our system

Dependency Parsing	
Annotation by Stanford	Name of the Function
ROOT	Predicate (core of sentence)
nsubj	nominal subject
nmod	nominal modifier
cop	copula
cc	coordinating conjunction
conj	conjunct
avdmod	adverbial modifier

```
tokenization: ['What', 'is', 'Genghis',
      'Khan', ''s', 'real', 'name', '?']

parsing:
(ROOT
  (SBARQ
    (WHNP (WP What))
    (SQ (VBZ is)
     (NP
     (NP (NNP Genghis) (NNP Kan) (POS 's))
     (JJ real) (NN name)))
    (. ?)))

dependency parsing:
[('ROOT',0,1),('cop',1,2),('compound',4,3)
,('nmod',7,4),('case',4,5),('amod',7,6),
('nsubj',1,7),('punct',1,8)]

NER:
[('What', 'O'),('is', 'O'),('Genghis',
'PERSON'), ('Khan', 'PERSON'),(''s', 'O'),
('real', 'O'),('name', 'O'),('?', 'O')]
```

4.2 Formalization of the rules into SPARQL

From Query Processing results, we establish a SPARQL request that queries the triple store. Parsing indicates that there is a noun phrase (NP) in the question. The latter composed of two proper names "Genghis" & "Khan". Dependency Parsing gives more information about the function of these words. Indeed, it tells us that "Genghis" is a compound word and his syntactic head is "Khan". Thus, these two proper names are linked together. Named entity recognition (NER) indicated the type of these two names which is "PERSON" type. Then, when we look back on dependency relationships, we see that the main kernel is "What".

Nevertheless, in dependency relationships, we can't have "ROOT" type of interrogative pronoun. In addition, "ROOT" is followed by a copula. In this case, according to the third rule, the intention of a question is symbolized by the nominal subject. The nominal subject in this question is "name" so it becomes "ROOT" of the question. Therefore, it is considered as the main predicate. These annotations are expressed in the following request:

```
SELECT DISTINCT ?reponse
WHERE {?x onto:name ?reponse.
       ?x rdf:type onto:Person.
       ?x rdfs:label 'Genghis Khan'.}
```

4.3 SPARQL into Answer

After transforming user's question into a SPARQL request, we query the triple store to get the answer corresponding to the question. In order to do this, we use a SPARQL Wrapper [LZB17] , it's a wrapper around a SPARQL service that allows us to query the URI of our triple store.

User: What is Genghis Khan's real name?
Bot: Temujin.

5 Evaluation

5.1 Background

To evaluate the performance of our approach, an ontology has been used that symbolizes the concept *person* Figure 3. The ontology represents personal information about a person x, such as date of birth, place of birth, profile description, job, etc. Therefore, the ontology contains classes (*Person*, *Organization*, *Occupation*, and *Location*, etc), object properties (*wasBorn*, *isLocated*, *hasOccupation*, etc) and data properties (*born*, *cost*, *description*, etc).

Precision, Recall and F-measure have been used to compare our approach with A. Bouziane et al [BDNM18]. The main purpose of this evaluation is to evaluate chatbot's ability to identify the user's intent. In order to do this, we submit a dataset of questions related to the ontology *person*. These questions were built by Yassine Benajiba [RBL06]. In this study case, the questions will be asked in order to extract personal information related to a person, his family, his job, etc.

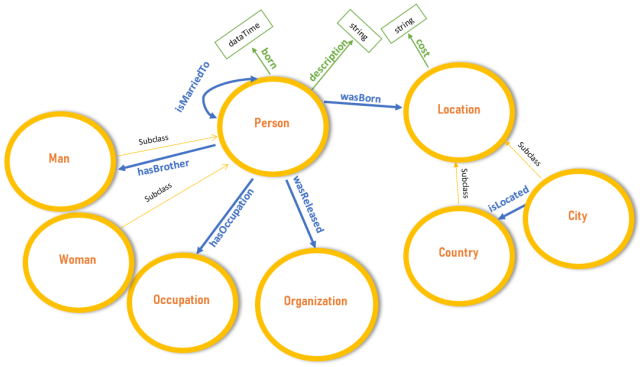


Figure 3: Ontology used during evaluations

$$Precision(P) = \frac{\text{Number of Correct Answer}}{\text{Total Number of Answer}}$$

$$Recall(R) = \frac{\text{Number of Correct Answer}}{\text{Total Number of Correct Answer}}$$

$$F{-}measure = 2 \times \frac{(\text{P x R})}{(\text{P} + \text{R})}$$

5.2 Results

A comparison was made between our system approach and an Arabic Question Answering System [BDNM18]. Their system attained respectively 0.71, 0.66, 0.68 for the precision, recall and F-measure. However, our system successfully achieves, as you can see in Table 2, 0.88, 0.86 and 0.87 in terms of Precision, Recall and F-measure.

The main challenges in our approach were the formalization of the rules and the building of the ontology. In fact, the rules are formalized manually and it requires significant corpus of questions to elaborate generic rules. These rules demand a permanent renewal as soon as specific cases arise. In addition, in order to deal with the questions present in the dataset, it is necessary to create a same domain ontology than [BDNM18]

Table 2: Evaluation results, expressed in Precision, Recall and F-measure.

Results		
Measure Performance	System Approach	A. Bouziane et al.
Precision	**0.88**	0.71
Recall	**0.86**	0.66
F-measure	**0.87**	0.68

with all the concepts, relations and instances. The implementation of these two things may take time but our method is fruitful because the results show that the system is very efficient to find the user's intent and they are much higher than [BDNM18] approach. This is because we have essentially used dependency relationships to understand user's intent.

Indeed, these relationships have helped us to understand not only the meaning of user's question but also the structure of his question. This linguistic information are then managed by the rules that we have formalized in order to be translated into triple patterns and find the answer to the question. While [BDNM18] put forward the stop words removal to identify the user's intent. This method is too drastic and not applicable to certain number of questions. In fact, by removing the empty words, this can affect or destroy the meaning and especially the structure of the question. Therefore, their system may provide an incorrect answer which clearly distorts the results of their approach.

We believe that our method represent a significant improvement of the state of the art QA systems due to the development of a generic method that deals with different cases and different ways that the questions were asked. Indeed, using the NLP methods helps us to identify user's intent, however there is still room for improvement. For example: dealing with several intents in a question. Actually, our system can only detect one intent at a time. Then, we can make automatic rules generation.

6 Conclusion & Future Works

One of the biggest challenges in the development of question answering system is to advance a conversational system or chatbot that is not based on preparation in upstream of questions and answers template. This paper presents a question answering system based on NLP methods and Semantic Web techniques to provide answers to questions expressed in natural language. In our approach, we use NLP methods to process user's question. In this processing, we use syntactic dependency relationships to view the semantic and syntactic structure of the question. These relationships are very important to understand and correctly answer user's question. Then, we transform them into SPARQL queries by means of rules in order to query our triple store. Indeed, the knowledge of the chatbot has been represented using an ontology. The evaluation of our approach shows that our method is good as our system can be adapted to a large number of questions. This shows that our approach constitutes a significant step in the question answering field.

However, the proposed approach requires improvement in future works. First, we will have to rework the intent of the bot. Indeed, the bot can respond and process one intention at a time for now. Secondly, the formalization of the rules is still manual and it will be better to take into account this issue. Then, we will introduce the ontology alignment in our method in order to deal with open domain and ameliorate results. Finally, we can customize our bot, by adding vocal conversations. This implies implementation of machine learning and advanced algorithms.

References

[A+18] A. ARAIN et al. Artificial intelligence mark-up language based written and spoken academic chatbots using natural language processing. *Sindh University Research Journal-Science Series*, 50:153–158, mar 2018.

[ABK+07] Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. volume 6, pages 722–735, 01 2007.

[AKS17] Ali Albarghothi, Feras Khater, and Khaled Shaalan. Arabic question answering using ontology. *Procedia Computer Science*, 117:183–191, dec 2017.

[APMG12] Agnese Augello, Giovanni Pilato, Alberto Mach, and Salvatore Gaglio. An approach to enhance chatbot semantic power and maintainability: Experiences within the frasi project. In *IEEE International Conference on Semantic Computing*, pages 186–193, sep 2012.

[BAB12] Raju Barskar, Gulfishan Ahmed, and Nepal Barskar. An approach for extracting exact answers to question answering (qa) system for english sentences. *Procedia Engineering*, 30:1187–1194, dec 2012.

[BC14] Ritika Bansal and Sonal Chawla. An approach for semantic information retrieval from ontology in computer science domain. *International Journal of Engineering and Advanced Technology*, 4:58–65, dec 2014.

[BDNM18] Abdelghani Bouziane, Bouchiha Djelloul, Doumi Noureddine, and Malki Mimoun. Toward an arabic question answering system over linked data. *Jordanian Journal of Computers and Information Technology*, may 2018.

[CLR13] L. Chiticariu, Yunyao Li, and F.R. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA, oct 2013.

[Cor19] Stanford CoreNLP Natural Language software, https://stanfordnlp.github.io/CoreNLP/index.html, 2019.

[Fer16] Sbastien Ferr. Sparklis: An expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8:405–418, dec 2016.

[KBPK17] Chaitrali S. Kulkarni, Amruta U. Bhavsar, Saviata R. Pingale, and Satish S. Kumbhar. Bank chat bot - an intelligent assistant system using nlp and machine learning. *International Research Journal of Engieering and Technology*, 4:2374–2376, may 2017.

[LZB17] Maxime Lefrançois, Antoine Zimmermann, and Noorani Bakerally. A SPARQL extension for generating RDF from heterogeneous formats. In *Proc. Extended Semantic Web Conference (ESWC'17)*, Portoroz, Slovenia, May 2017.

[Mac18] Machinalis. Quepy's project documentation, 2018.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995.

[NAJ14] Lucas Fonseca Navarro, Ana Paula Appel, and Estevam Rafael Hruschka Junior. Graphdb – storing large graphs on secondary memory. In *New Trends in Databases and Information Systems*, pages 177–186, Cham, 2014. Springer International Publishing.

[NNBU+13] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. Sorry, i don't speak sparql: Translating sparql queries into natural language. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 977–988. ACM, 2013.

[RBL06] Paolo Rosso, Yassine Benajiba, and Abdelouahid Lyhyaoui. Towards an arabic question answering system. *Proc. of SRO4*, jan 2006.

[SWRR14] Malte Sander, Ulli Waltinger, Mikhail Roshchin, and Thomas A. Runkler. Ontology-based translation of natural language queries to sparql. In *AAAI Fall Symposia*, 2014.

[YD14a] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 1, pages 956–966, jun 2014.

[YD14b] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 1, pages 956–966, jun 2014.

Auto-Perceptive Reinforcement Learning (APRiL)

Rebecca Allday, Simon Hadfield, and Richard Bowden
Center for Vision, Speech and Signal Processing (CVSSP)
University of Surrey, Guildford, United Kingdom
{r.allday, s.hadfield, r.bowden}@surrey.ac.uk

Abstract

The relationship between the feedback given in Reinforcement Learning (RL) and visual data input is often extremely complex. Given this, expecting a single system trained end-to-end to learn both how to perceive and interact with its environment is unrealistic for complex domains. In this paper we propose Auto-Perceptive Reinforcement Learning (APRiL), separating the perception and the control elements of the task. This method uses an auto-perceptive network to encode a feature space. The feature space may explicitly encode available knowledge from the semantically understood state space but the network is also free to encode unanticipated auxiliary data. By decoupling visual perception from the RL process, APRiL can make use of techniques shown to improve performance and efficiency of RL training, which are often difficult to apply directly with a visual input. We present results showing that APRiL is effective in tasks where the semantically understood state space is known. We also demonstrate that allowing the feature space to learn auxiliary information, allows it to use the visual perception system to improve performance by approximately 30%. We also show that maintaining some level of semantics in the encoded state, which can then make use of state-of-the art RL techniques, saves around 75% of the time that would be used to collect simulation examples.

1 Introduction

Unifying deep reinforcement learning with visual perception is often slow and ineffective for high dimensional problems with continuous action spaces. This is perhaps unsurprising as training directly from percepts through to actions for control is a complex relationship with poorly constrained supervision. This is especially true due to the high dimensionality of the image domain and the fact that many techniques used to speed up learning cannot be applied in image based systems. In nature, learning to interpret our surroundings and interact with them are learnt simultaneously but not necessarily as one continuous system [1]. Inspired by this, we use interaction with the world to co-train separate visual perception and control systems (Fig. 1).

This work shows how an auto-perception network can be used to learn an effective state space for reinforcement learning. Unlike previous RL techniques which try to learn an encoded state space, the proposed solution generalises across all levels of state observability. When the state space is completely or partially observable at training time then it is used to condition the learning of a representative feature space. The conditioned auto-encoder can be used to create a feature space which includes this knowledge but is not limited to it - allowing retrieval of other auxiliary information from the observations which may not have been considered by the developer.

Copyright © by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

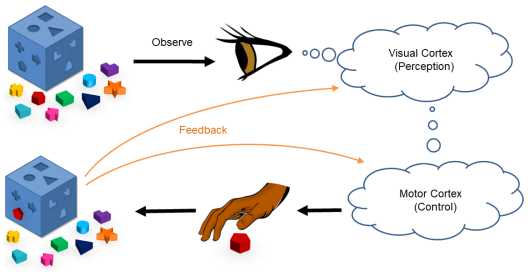


Figure 1: Perception and control in the brain are linked but separate systems requiring different feedback.

2 Related Work

Deep RL has seen advances recently with work like Deep Q-Networks [2] which uses a deep convolutional neural network (CNN) to approximate the action-value function in a Q-learning method to learn to play Atari games. There have since been many variations on DQNs such as using recurrent neural networks in place of a standard feed-forward CNN [3] and adaptations for use with continuous action spaces [4]. Whilst these value based methods for RL have proved popular, policy based and actor-critic methods have also been successfully adapted for deep learning. In this work we use a synchronous version of Mnih et al.'s Asynchronous Advantage Actor-Critic (A3C) method [5].

RL algorithms are often tested using simple software simulators such as video games or simple physics problems (e.g. cart-pole). This makes it easy to accumulate the number of episodes required to train the networks, which is not practical for more realistic robotics applications. Many techniques for approaching the issue of data collection have been suggested. For example, Hindsight Experience Replay (HER) [6] allows RL to learn from unsuccessful episodes by changing the goal and hence the reward feedback. However, in order to apply this to the image domain, a method for synthesising images is required to change the goal. There have also been model based techniques aimed at reducing the number of experiences needed for training. Black-DROPS (Black-box Data-efficient RObot Policy Search) [7], for example, uses Gaussian Processes (GPs) to learn the dynamics of a system with a small number of experiences and then produces experiences for training the RL directly from the GPs. This accelerates the RL process but is focused on systems where the state is fully observed and has a small number of dimensions. The large dimensionality of observations only available as images are not suitable for GP dynamics modeling.

Advances in deep learning has meant that feature spaces can be created which represent the important aspects of a visual observation. Deep auto-encoders [8] have been used extensively to reduce dimensionality of data and have been used with CNNs [9] to help retain the spatial relationships in images. As well as providing a low-dimensional feature space they are also used to create generative models, for example in image restoration [10].

Considering the problems high dimensional spaces cause in RL, it is not surprising that attempts to use auto-encoder networks with RL have been made. Table 1 compares the uses of auto-encoders in RL systems. Finn et al. [13] use an auto-encoder to create a set of feature points representing positions in the image that describe the environment, for example where objects are. Stadie et al. [15] encode a state for training a dynamics model in order to improve exploration by increasing curiosity, but still use the raw observation as the input to the learning system. Lange and Riedmiller [11] use a deep auto-encoder to compress a visual input to a low dimensional feature space, which is not semantically understood. This improves the reinforcement learning data-efficiency. Kimura [16] uses auto-encoders as pre-training for a DQN system. However, none of these approaches can exploit valuable RL techniques, such as HER. Lange and Reidmiller's work does not have the semantic understanding required in the features in order to adapt the episode with a new goal. Kimura's requires images, for fine-tuning of the network, which cannot be adapted for a new goal.

Nair et al. [17] propose a solution to goal-conditioned RL, using an encoder-decoder system to learn a latent space which can be used to sample goals, provide a lower dimensional, structured input for RL, and to compute a reward signal. Although this allows HER to be used for visual problems, it introduces its own limitations. In using an image as an explicit goal, the agent's flexibility is limited. For example, in a pick and place problem it restrains the final position of the robot when the final position of the object is more important. They also assume that only the image is available to the RL system at train time, they do not consider cases where we may want to make use of the state that is available - meaning information is wasted.

In contrast APRiL makes use of whatever semantically understood state information is available at train time,

Table 1: Comparison of different works using Reinforcement Learning (RL) and Auto-encoders (AE)

	Use of AE	**RL method**	**RL Input Space**	**Goal conditioned**	**Semantics in RL input**
Lange and Riedmiller [11]	Encode image	FQI [12]	Latent space	No	None
Finn et al. [13]	Encode image	Gaussian Controller + Guided Policy Search [14]	Robot state + latent space	Implicit in image - encoded into latent space	Partially
Stadie et al. [15]	Encode image for augmented reward	DQN [2]	Images	Implicit in image	-
Kimura [16]	Pretrain RL network	DQN [2]	Images	Implicit in image	-
Nair et al. [17]	Encode image, Calc reward, Generate data	TD3 [18]	Latent space	Conditioned with a point in the state space	None
APRiL (Ours)	Encode image	A2C [5]	Available env and robot state + latent space	Implicit in input (state or latent space)	Variable

whilst still allowing additional auxiliary information to be encoded from the visual input. This gives a system which makes full use of the information and RL techniques available at train time but can still be deployed using vision as the input.

3 Methodology

Fig. 2 shows the outline of the proposed APRiL approach. The black arrows show the data flow at deployment. The observation image of the agent and environment is passed to the encoder which provides an encoded state, which may be completely or partially semantically understood. This is then passed to the trained reinforcement learning system which selects an action which is passed back to the agent to be executed.

The flow of the data when the system is being trained can also be seen in Fig. 2. The optional loss can be used if semantically understood knowledge of the state is partly or fully available. The perception network is trained independently on data collected with an initial random walk policy from the RL system. The RL block is trained using data both from the agent (in this case a physics simulator) and from a pre-trained Gaussian Process which models the dynamics of the system. This means that the RL system can obtain vast quantities of data points without having to run them all through a physics simulator, speeding up the process. The following sections elaborate on the individual elements and how they are trained.

3.1 Reinforcement Learning

A formalisation of episodic reinforcement learning is used where an agent interacts with an environment at discrete time steps, t, with a maximum number of steps T. There is a set of states $s_t \in \mathcal{S}$ and a set of actions the agent can perform $a_t \in \mathcal{A}$. The goal is to maximise the discounted sum of reward signal r_t over time,

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{1}$$

where $\gamma \in [0, 1]$ is the discount factor for future rewards. In order to maximise R_t we learn a policy $\pi(a \mid s_t)$, which estimates a distribution over the possible actions, $a \in \mathcal{A}$, conditioned on the current state s_t. We sample a_t from this distribution $\pi(a \mid s_t)$. The value is defined as $V^\pi(s_t) = \mathbb{E}\left(R_t \mid s_t, \pi\right)$, the expected return R_t given a particular policy starting in a particular state s_t. Finally, the action-value function is defined as $Q^\pi(s_t, a_t) = \mathbb{E}\left(R_t \mid s_t, a_t, \pi\right)$,

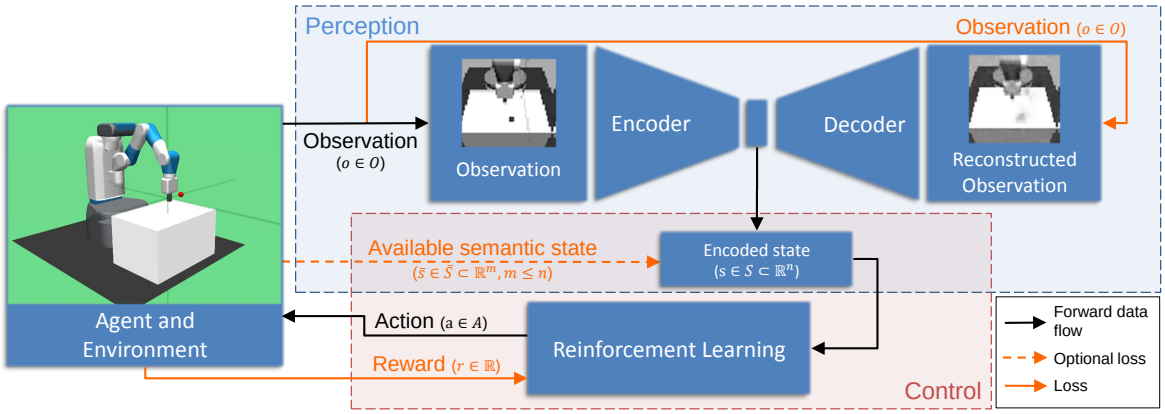


Figure 2: Overview of APRiL. The optional loss and the $|\mathcal{S}|$ determines how much of the encoded latent space is semantically understood. Black arrows: the data flow in the forward pass, Orange arrows: the data flow in the backward pass.

the expected return R_t given a particular policy, starting with a particular action a_t from a specified stated s_t. For the visual aspect we define $o_t \in \mathcal{O}$ as an image of the system.

In this work we use Advantage-Actor-Critic style reinforcement learning [5]. This system produces two outputs - a stochastic policy (the actor) and an estimate of the value function (the critic). The ground truth value R_t is used to calculate the value loss

$$L_v = R_t - V(s_t). \tag{2}$$

The policy loss is calculated using the advantage, given by

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t). \tag{3}$$

The advantage gives the difference between the expected return given the action taken and the expected return of the state itself given the current policy - showing how much better or worse the action performed than expected. This can be approximated as the discounted rewards minus the predicted value for the current policy, taking the form $A(s_t, a_t) \approx R_t - V(s_t)$. The policy used is in the form of a Gaussian distribution, such that $\pi(a \mid s_t) = \mathcal{N}(\mu_a, \sigma_a^2)$. Given that an action a_t is then sampled and executed, the policy loss is then calculated as

$$L_p = \log \pi(a = a_t | s_t) A(s_t, a_t). \tag{4}$$

This means that an action which is better than expected will be made more likely, with a weighting of how likely it was in the first place. In contrast an action which performed worse will be made less likely for that state. The full loss for the RL network then takes the form

$$L_{\text{RL}} = \alpha L_v + \beta L_p + \epsilon H(\pi(s_t)) \tag{5}$$

where H is the entropy - which is included to encourage exploration - and α, β, ϵ are hyper-parameters which control the strength of each loss term. To ensure that the initial random value estimate is sensible and does not skew the policy loss, we train with $\alpha = 1, \beta = 0, \epsilon = 0$ for a small number of iterations.

We use a batch-style off-policy approach by storing up experience in a replay buffer and sampling from this to train the RL algorithm. We set a limit to our experience replay buffer to some value M so that as learning progresses, the oldest experiences are forgotten and replaced with more recent ones. The replay buffer is of the form $\Omega = \{e : |\Omega| < M\}$, where each episode of experiences is of the form $e = \{(s_t, a_t, R_t) : t = 1, .., j \text{ and } j \leq T\}$ where j is the terminating step for that episode. The probability of a_t being selected from the current policy and the value of the s_t for the current policy is found at training time.

3.1.1 Hindsight Experience Replay

Hindsight Experience Replay (HER) [6] is a powerful technique which allows us to learn from unsuccessful episodes in learning, especially where rewards are sparse and success from random exploration may be limited. Using HER we can adjust the goal for our system to a state it achieved in the current episode - meaning we artificially

create successful episodes. For a given episode $s_1, ..., s_T$ where a goal $g \neq s_1, ..., s_t$, we may "replay" this episode with $g = s_i$ for some $1 < i < T$ knowing that it will achieve the goal. Adding these adapted episodes to the experience replay, Ω, means the episode buffer then has more episodes to learn from and has a more balanced ratio of successful episodes without needing excessive exploration.

3.1.2 Gaussian Process Model

In order to reduce the number of costly agent-environment interactions we use Gaussian Processes (GPs) to approximate the dynamics of our system and give uncertainty information. We use a small number of interactions with the agent and environment to train the GP - this takes in the current state, s_t, and the action to be taken, a_t. It is then optimized to output a Gaussian distribution which estimates the next state s_{t+1} with uncertainty.

We represent the dynamics of our system as

$$s_{t+1} = s_t + D(s_t, a_t) + \mathrm{w}, \tag{6}$$

with w (Gaussian system noise) and D (unknown transition dynamics). Given that $x_t = (s_t, a_t)$, the GP is computed as

$$\hat{D}(x_t) \sim \mathcal{GP}(\mu_{\hat{\mathrm{D}}}(x_t), k_{\hat{\mathrm{D}}}(x_t, x_t')), \tag{7}$$

where $\mu_{\hat{\mathrm{D}}}$ is the mean function and $k_{\hat{\mathrm{D}}}$ is the kernel function. With a set of observed transitions $Y_{1:t} = D(x_1), ..., D(x_t)$, we can query our GP at a new data point x_* to obtain a distribution over expected state updates:

$$p(\hat{D}(x_*) \mid Y_{1:t}, x_*) = \mathcal{N}(\mu_{\hat{\mathrm{D}}}(x_*), \sigma^2_{\hat{\mathrm{D}}}(x_*)). \tag{8}$$

Sampling from this Gaussian allows the rapid creation of more episodes to train the RL system. The same reward calculations as the normal environment are used so these episodes can be added directly to Ω as before.

3.2 Auto-Perceptive Network

The perception part of our system is an auto-encoder. This allows us to encode a feature space to use as the state space, $\mathcal{S}$, which is the input to the RL system. The encoder uses the observations of the agent and environment in the form of an image, transforming it to the feature space as the function $\phi_{enc} : \mathcal{O} \rightarrow \mathcal{S}$, whilst the decoder arm transforms from the feature space to a reconstructed image $\phi_{dec} : \mathcal{S} \rightarrow \mathcal{O}$.

The auto-encoder takes the image observation of the system as an input and compresses it down to the feature space $s_t = \phi_{enc}(o_t)$ and the output is a reconstruction of that image $\hat{o}_t = \phi_{dec} \circ \phi_{enc}(o_t)$. The reconstruction loss is a pixel-wise loss against the input

$$L_r = |o_t - \hat{o}_t|. \tag{9}$$

We denote the space of available information from the environment, which has a predefined semantic meaning, as $\bar{s}_t \in \bar{\mathcal{S}}$. The optional conditioning loss is the absolute difference between a section of the encoded state space and the semantically understood state. In the case where $\mathcal{S} \subset \mathbb{R}^n$ and $\bar{\mathcal{S}} \subset \mathbb{R}^m$, with $m \leq n$, then the conditioning loss is

$$L_c = |s_t^{1:m} - \bar{s}_t|. \tag{10}$$

The full loss for the visual perception network is

$$L_{\mathrm{VP}} = L_r + \omega L_c, \tag{11}$$

where ω is a weighting which determines how strong the conditioning is. The learnt feature space can be:

1. entirely conditioned to be semantically understood as the observable state ($m = n$, $\omega \neq 0$),
2. partially conditioned with some learnt features relating to the observable state and some auxiliary features with no predetermined semantic meaning ($m < n$, $\omega \neq 0$),
3. or not conditioned with learnt features having no predetermined semantic meaning ($\omega = 0$).

This network can be trained using data from initial random exploration and fine-tuned during reinforcement learning.

3.3 Auto-Perceptive Reinforcement Learning (APRiL)

The RL system and the auto-perception network are independent networks, which can be trained concurrently with much of the same data but do not need to be trained end-to-end as they exploit different types of supervision.

In the case of the encoded feature space being entirely semantically understood the auto-encoder is trained with data collected for the initial random exploration - the same data can be used to train the GP to learn the dynamics. These may be co-trained in parallel and tested individually before being integrated. The perception network infers an approximated state from an observation and then passes this approximate state to the RL network without the RL needing to see any images during training. This still provides a system which does not need access to the robot state at run time and can predict actions with only visual input, but does not require it to be trained in an end-to-end manner, allowing RL to benefit from HER and GP modelled transition dynamics.

When the encoded feature space is partially semantically understood then the auto-encoder will still be pre-trained on random data but the encoder arm will be used to get the encoded state for input to the RL system. Therefore the RL system only has to interpret the low dimensional feature space coming from the auto-encoder and does not need to process the images. This means that the training is more focused on solving the control problem. Techniques such as HER are still feasible since we have a predetermined understanding of some of the feature space being used by the RL.

The final case is where there is no semantically understood state available. This is similar to Lange and Riedmiller's work [11] where the encoder feature space had no predetermined semantic meaning. This case still allows a lower dimensional state space to be learnt from the visual input even when there is no semantic state available during training.

4 Experiments and Results

To evaluate APRiL we use the OpenAI [19] framework with the Mujoco physics simulator [20]. We use a variation of the Fetch robot reach environment because it has a continuous action space and has a visually interesting environment to test the auto-perceptive system. The aim is to direct the end-effector of the Fetch arm to a goal g_x, g_y, g_z - represented visually by a red sphere. The action space is defined with actions $(\Delta x, \Delta y, \Delta z)$ where (x, y, z) is the position of the end-effector and the maximum episode length is set as $T = 50$. We train the networks using Adam optimizers [21] and a Tensorflow [22] implementation of our system will be available at `https://github.com/rebecca-allday/APRiL`.

4.1 Fully Semantic Features

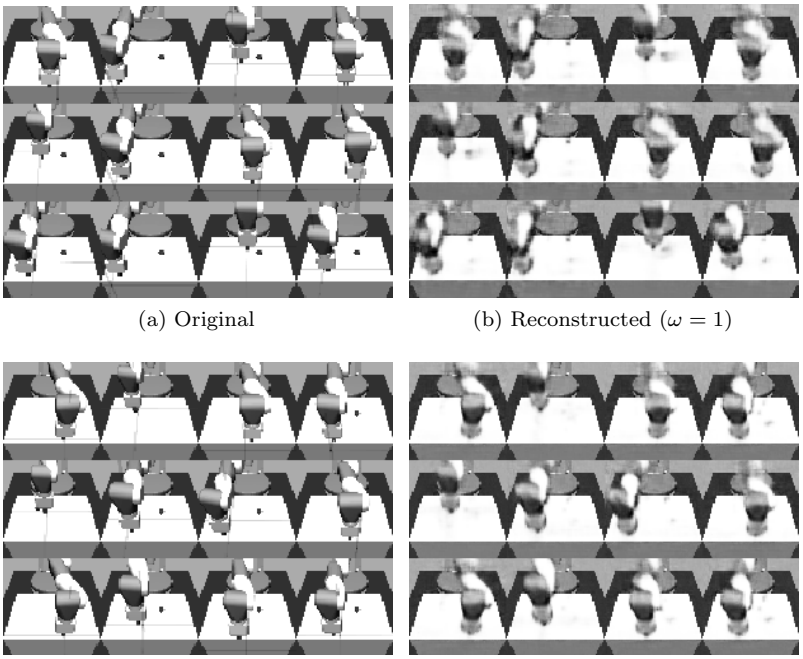
The first experiment uses a fully observed, semantically understood state $\bar{s}_t = (x_t, y_t, z_t, g_x, g_y, g_z)$. Firstly, we use a random policy to collect an initial experience replay buffer. This data can be used to train multiple aspects of the system. Initially we train a GP on the transitions taking in $(x_t, y_t, z_t, \Delta x, \Delta y, \Delta z)$ and outputting $(x_{t+1}, y_{t+1}, z_{t+1})$. This allows us to create extra episodes to train our RL system as described in Section 3.1.2. The advantage actor-critic RL system is trained with data created from both the GP and from the agent, including the HER additions to the replay buffer. The data from the random policy and any episodes collected using the simulator are used to train the perception network. In this case the perception network is co-trained such that $\bar{s}_t = s_t = \phi_{enc}(o_t)$, which is the first case from Section 3.2, when $m = n$ and $\omega \neq 0$. Finally, at test time the networks can be used together to go directly from vision to actions, following the data flow shown by the black arrows in Fig. 2. We compare this to a latent space with no conditioning loss, where $\omega = 0$, which is similar to [11].

The training of the RL system, using the semantically understood state space directly, converges with only 15 episodes of random policy interactions with the simulator, the rest of the data used is collected from our trained GP. It takes approximately 0.01 seconds per rendered simulation step, but only 0.0025 seconds to sample a single step from the GP. This equates to saving 75% of the time that would have been spent on collecting simulation examples. This is a saving that would not be possible using a traditional end-to-end visual RL algorithm.

Table 2 shows the policy achieves an average episode length of 3.1 actions when using the ground truth state space as input. The perception network is trained alongside this. Examples of the reconstructions from the auto-encoder can be seen in Fig. 3, along with reconstructions from the auto-encoder without the semantic conditioning ($\omega = 0$). Even though we fully constrain the encoded feature space, and do not enable the system to encode many visual properties, the decoder arm is still able to learn how to produce realistic images of the scene from a non-visual intermediate state, including how to correctly place a fully textured robotic arm. They

Table 2: Average episode length (actions to complete task) of system trained on the Fetch Reach env (no obstacle).

Runtime RL Input	Average Episode Length
Ground truth $\bar{\mathcal{S}}$	3.10
Perceived $\mathcal{S}$ ($\omega = 1.0$)	12.42
Perceived $\mathcal{S}$ ($\omega = 0.5$)	17.06
Perceived $\mathcal{S}$ ($\omega = 0.0$) [11]	30.94



(a) Original (b) Reconstructed ($\omega = 1$)

(c) Original (d) Reconstructed ($\omega = 0$)

Figure 3: Reconstructions from the auto-perceptive network

are certainly comparable to the reconstructions without the conditioning loss. However, reconstruction accuracy is unimportant, the key is the reconstruction loss aids encoding meaningful information into the latent space for the RL.

At test time we can see the performance of the system using the visual encoder network to produce the feature space, which is an approximation of the semantically understood state space, given to the RL network. The policy achieves an average episode length of 12.4 actions. This is largely due to the goal or end point being occluded or out of the field of view, in which case the arm must move to attempt to gather more information about its current state. In these situations, the ground truth algorithm is an unrealistic comparison for a vision based system which will never have full access to the state. However, this is still much more effective than the case when the perceived state, $\mathcal{S}$, is not conditioned on the semantically understood state, $\bar{\mathcal{S}}$, which is similar to [11].

4.2 Partially Semantic Features

The next set of experiments introduces an element such that the state is not be fully observed via a semantically understood state space. A randomly placed obstacle (box) is added which can affect exploration and potential solutions for getting to the goal (red sphere), see Fig. 4. Again we compare the results in this section to a network trained with no access to the available semantic state which is similar to [11]. We also train a system which takes the ground truth semantic state and a separate latent space (in a similar way to [13]) to show that if both are available the system has all the information it needs.

We first train APRiL on the same state that was available in the previous set-up. This means that the RL

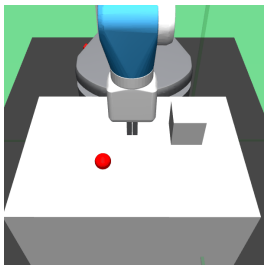


Figure 4: Fetch simulation with obstacle (box) and goal (sphere)

Table 3: Average episode length of system trained on an environment with a randomly placed obstacle.

Runtime RL Input	n	m	Average Episode Length
Ground Truth $\bar{\mathcal{S}}$	-	-	8.45
GT $\bar{\mathcal{S}}$ and Percieved $\mathcal{S}$ [13]	6	0	5.44
Perceived $\mathcal{S}$ [11]	16	0	37.04
Perceived $\mathcal{S}$	6	6	28.55
Perceived $\mathcal{S}$	8	6	20.83

system is not receiving any information about the obstacle. As expected, we see a reduction in performance compared to the environment with no obstacle. From 3.10 average actions per episode with no obstacle to 8.45 with obstacles - this equates to approximately a 2.5 times increase in the number of actions. Examples of the reconstructions from the perception network are seen in Fig. 5b. These reconstructions are comparable to those in Fig. 3b, with some slight degradation because the scene is more complex yet we have not given it any additional degrees of freedom in the latent space. It is interesting to note that the decoder arm attempts to reconstruct the obstacle even though it is theoretically not present in the intermediate state.

When testing with the perception to action system we see that this gives much worse performance with an average episode length of 28.55 actions (See Table 3). It is good to note that this is in comparison to 12.42 actions with no obstacles, equating to approximately a 2.5 times increase in the number of actions which is similar in scale to the decrease in performance seen without perception. This is likely because it has no way of knowing about the obstacle in the encoded state and often mistakes it for the goal, especially if the goal is occluded by the arm.

Next we allowed the encoded feature space to be only partially semantically understood. We used a feature space of size $n = 8$, with the semantically understood state $\bar{s}_t$ conditioning only the first 6 elements (i.e. $m = 6$). The rest were driven purely by the reconstruction loss, allowing it to learn whatever was relevant to the understanding of the environment. Example reconstructions from the perception network can be seen in Fig. 5d. This trained perception network does a better job of modelling the obstacle and goal as independent objects, however the robot arm has lost a significant amount of visual fidelity. This may be because all systems have been trained for the same number of iterations, despite this one having more network parameters. Regardless, a high fidelity image of the robotic arm is not important for RL, as long as the position is known.

The proposed RL system using our partially semantically understood feature space as input performs better than the system using just the semantic state, with an average of 20.83 actions (See Table 3). In comparison to the 12.42 actions in the environment with no obstacles, this is only a 1.68 times increase for what is a more difficult problem. This is approximately a 30% improvement compared to 28.55 average actions taken when using the semantic feature space. This shows that when we do not have access to the full semantically understood state our feature space can encode the additional auxiliary information necessary to solve the task better than just with the semantic state based perception.

Finally we give the perception network complete freedom to encode a state space based purely on the reconstruction loss in a similar manner to [11]. Fig. 5f shows that this improves the reconstruction as expected since that is the only feedback given to the encoder-decoder network. However, as we can see from Table 3 the performance of the system with no use of the semantically understood data available to it at train time performs much worse than those which do.

(a) Original (b) Reconstructed ($n = m = 6$)

(c) Original (d) Reconstructed ($n = 8, m = 6$)

(e) Original (f) Reconstructed ($n = 16, m = 0$)

Figure 5: Reconstructions from the auto-perceptive network for the env with obstacles - top: semantic features, middle: partially semantic features, bottom: non-semantic features.

5 Conclusion

In this paper we have shown that the bio-inspired separation of percepts and control at training time allows reinforcement learning to be trained effectively and still gives a system that can predict actions purely from visual data. We showed that allowing the perception system to encode additional properties into the feature space improved the performance over a system using only the approximate state.

This demonstrates the value in allowing the visual system to encode additional features into the input of our RL algorithms. In addition, the splitting of perception and control allows other techniques to be used, which are typically challenging to implement in the high dimensional image domain, such as HER and modelling transition dynamics with GPs. Whilst we still have a system which allows us to go from visual observation to action - the training does not need to be end-to-end.

References

[1] M. Land and B. Tatler, *Looking and acting: vision and eye movements in natural behaviour.* Oxford University Press, 2009.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, 2015.

[3] M. Hausknecht and P. Stone, "Deep Recurrent Q-Learning for Partially Observable MDPs," *AAAI*, pp. 29–37, 2015.

[4] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, pp. 1–14, 2015.

[5] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML*, 2016, pp. 1928–1937.

[6] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," *CoRR*, vol. abs/1707.01495, 2017.

[7] K. I. Chatzilygeroudis, R. Rama, R. Kaushik, D. Goepp, V. Vassiliades, and J. Mouret, "Black-box data-efficient policy search for robotics," *CoRR*, vol. abs/1703.07261, 2017.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504–507, 2006.

[9] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks.* Springer, 2011, pp. 52–59.

[10] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems 29.* Curran Associates, Inc., 2016, pp. 2802–2810.

[11] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, July 2010, pp. 1–8.

[12] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 503–556, 2005.

[13] C. Finn, X. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *ICRA*, 2016.

[14] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, 2014, pp. 1071–1079.

[15] B. C. Stadie, S. Levine, and P. Abbeel, "Incentivizing exploration in reinforcement learning with deep predictive models," *CoRR*, vol. abs/1507.00814, 2015.

[16] D. Kimura, "DAQN: Deep Auto-encoder and Q-Network," *arXiv*, vol. abs/1710.06542, 2018.

[17] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," in *Advances in Neural Information Processing Systems*, 2018, pp. 9209–9220.

[18] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv:1802.09477*, 2018.

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[20] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on.* IEEE, 2012, pp. 5026–5033.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.