Team Description Paper of TJArk@Home

He Zongtao
Robot and Artificial Intelligence Lab
Tongji University
1930719@tongji.edu.cn

Zhou Xun
Robot and Artificial Intelligence Lab
Tongji University
1930719@tongji.edu.cn

Xu Weihan
Robot and Artificial Intelligence Lab
Tongji University
1930719@tongji.edu.cn

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.

1 Introduction

As development of robotic technology, more and more people expect intelligent robots to serve the family as human beings and provide a good and comfortable life experience. The RoboCup@Home competition [1] focus on bringing robotics to apply in common domestic environment. To achieve such goal, the robotic system must be able to perceive environment, communicate with humans and complete various tasks, such as cleaning up a table, holding a party or guiding new comers. Security is the top priority. Because service robots can't do harm to people and the environment. Secondly, the robot should be controllable. The operator can intervene the behavior of the robot at any time to prevent the unexpected situation. Finally, the robot should have complete autonomous action ability and give people comfortable interaction experience.

The Robotics and Artificial Intelligence Lab (RAIL) of tongji university was founded in 1992, Team TJArk of RAIL was founded in 2004 and participated in RoboCup World Cup from 2006

to 2018. We have got seven winning streak in China RoboCup SPL and once won the third place in RoboCup2018 SPL. In December 2018, we founded an energetic new team to participate in RoboCup @Home League, called TJArk@Home. Main goal of TJArk@Home is to explore the limit of how well can robots serve people during common life with acceptable cost. Under the guidance of such wish, we are doing many research on the Pepper platform, such as

- SLAM(Simultaneous Localizaiton and Mapping)
- Auto navigation of robot
- Object Detection and Recognition
- Human-face detection and analysis
- Human-Robot interaction
- Motion control
- Trajectory teaching
- Design of laptop GUI
- ...

April 2019, the first time TJArk@Home involved in China RoboCup@Home League, we got the first-place in SSPL(Social Standard Platform League). During the competion, we show a lot of abilities, including Autonomous navigation, Follow an operator to the given position, Recognize speech and response, Detect humans in vision and summarize their features.

Although our team has not been established for a long time, we have developed a reusable and extensible software framework. It is a higher level integration of NAOqi API, which makes developing a new application more easily. The HTTP-based Information Exchange System(HIES) also helps a lot. Tasks require large computational resources such as syntactic analysis can be allocated to a host server. Pepper just need seed raw data and listen for response. Such mechanism reduce Pepper's burden, allowing it pay more attention to interaction logic and task excution.

Tongji RAIL lab has been researching on robotics since its begining, and achieved a lot. ??. RoboCup@Home League is a new attempt, which involves former research ??, but bringing untouched challenges ??. We are pushing researchs on vision, navigation, interaction and building a simulated competition scenario for integration testing.

2 Architecture

We have developed a Python framework, which is a high-level package of NAOqi APIs and ROS-based components. (??)

2.1 Native software on Pepper

On NAOqi OS, several processes are simultaneously running, including pad process, HTTP server/client process, motion control process, vision process, navigation process, etc. All codes are organized into two main part: libraries and skills. Libraries, we name them with "/Lib.*ṗy/" pattern, are files that contain various class definitions for basic ablilities. Every "Lib" attaches to some specific functions. For example, "LibVariables" defines a class related to global variables stored on local disk or remote server. "Lib"s are just useful tools without any task-oriented purpose.

To combine them together and implement specific applications, we write many app files, called "Skill". A "Skill" imports several "Lib"s, to accomplish a complete task, such as tracking a person. Every "Skill" runs as an individual process, so communication problems follows. OS's memory manager allocate different storage space for different process, so there is no native global space for our different "Skill" and "Lib". But most of time, there is a large need for them to share information. To overcome such dilemma, we developed a JSON-based global variable mechanism. As mentioned before, this function is implemented in "LibVariables". It uses hard-disk to store a global variable with JSON format. Thanks to separation principle of interface and implementation, the global space can be not only Pepper local disk, but also remote server memory/disk.

Overall, though many programs engaged, we just need to run a single shell script to start all applications we need. This native software system on Pepper is easy to learn, easy to use and easy to extend. Every team member can write his/her own lib files, such as "LibMoiton", "LibDetection", and add it to the framework by pushing their files to Pepper. The development process seems like building. It starts from a basic frame, and grows with bricks added by team members, and finally become a useful building.

2.2 HTTP-based Information Exchange System

Computational resources are limited in NAOqi OS. The practical requirement is that we must separate some computing tasks from Pepper and put them on a powerful remote server for execution. So, we designed a HTTP-based information exchange system. There are two HTTP server process, one in Pepper, another in external computer.

Main function of external computer server process is to listen Pepper's requests, start corresponding computing program and response the result to Pepper. A conspicuous example is speech recognition task. When records a complex speech sentence from the operator, Pepper send the audio file to the external computer through HTTP request. The server computer receives this request, and convert this speech to text. Finally, speech text is sent back carried by HTTP response package.

One the other hand, HTTP server process runs on Pepper mainly aims at web service. It contains a web site backend. When a browser login the URL, it gets a home page of various applications and can start them by simple clicks. Morever, a device just need the ability of sending HTTP requests to control Pepper's behavior. This part will be described at ?? in detail.

The advatage of HTTP information exchange is obvious, it is a widely used and device/language independent. But network quality would limit the performance of HTTP communication. Future works contain compressing information to reduce transfer load.

2.3 Robot Operation System (ROS)

Implementing SLAM and auto-navigation of robot needs well fusion of multi-sensor data and feedback control of motion actuators, so Robot Operating System (ROS) is a useful tool. We develop a basic module serving as a bridge between NAOQi and ROS, it publish Pepper's sensor data in ROS format and enable ROS to call NAOQi's APIs. ROS components get multi-sensor data, then do SLAM and path planning with real-time obstacle avoidance. As a result, motion commands will be sent to NAOQi.

3 Vision Simultaneous Localization and Mapping (VSLAM)

Pepper needs a model (a map) of the environment to support other tasks, mainly auto-navigation, but a prior map is often not available in various home scenarios, so SLAM find its application. The stereo camera data is mainly used by a visual SLAM module, together with laser (very limited) and odom (has significant error) data. Our VSLAM algorithm is based on RTAB-Map, a graph-optimized based VSLAM algorithm, and have been modified for Pepper's sensors. Odom and laser scan data make the initial pose estimation, then stereo data serves for appearance-based loop closure detection to optimize local and global pose. VSLAM can get a 3D point cloud, we make a projection to get a 2D grid map for navigation. Having map and Pepper's size parameters, we use DWA (Dynamic Window Apporach) to plan the path and fuse vision, laser and sonar data to make real-time obstacle avoidance (mainly for dynamic scene), a series of motion command will be sent to NAOqi APIs, so Pepper can navigate to its goal points itself.

4 Computer Vision

Various vision algorithms are needed for more careful perception and analysis of Pepper's environment pn object and huamn level. To start interaction, our Pepper can detect and recognize different people, remember their names and some other features, and track them using its camera data. Then, more interested skills will be shown to carry more complex tasks. For example, some tasks require

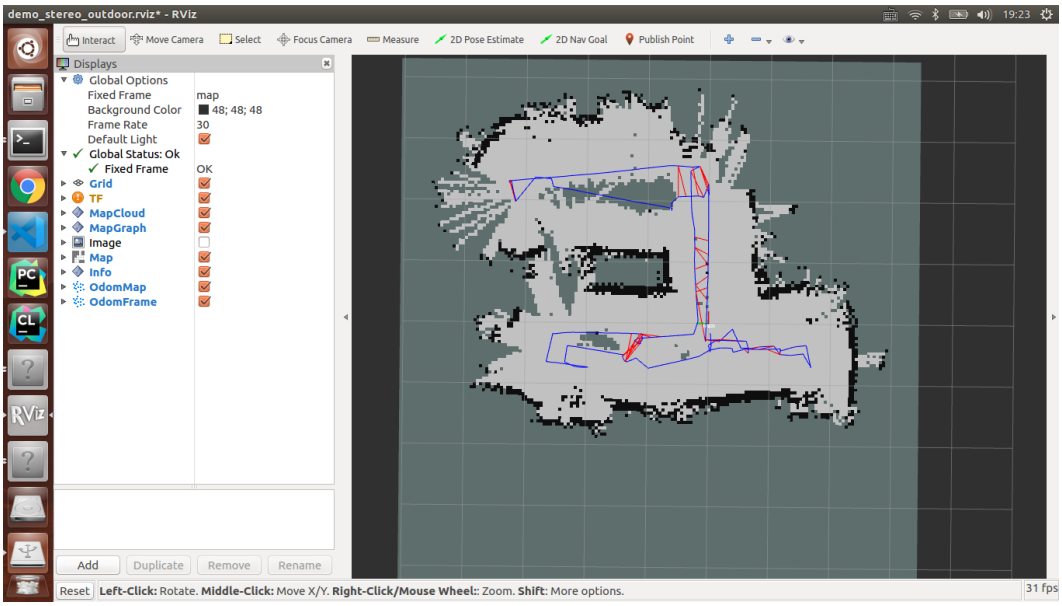


Figure 1: SLAM and navigation on ROS

human posture recognition so an alogorithm based on ?? is developed. It can recognize human skeleton so that we can judge whether a person is standing, sitting or pointing somewhere. For object-level environment reasoning, a detection and recognition framework based on YOLO can quickly detect a predefined object and print its predefined name in the figure. (??)

5 Human-Robot Interaction

5.1 Graphics User Interface

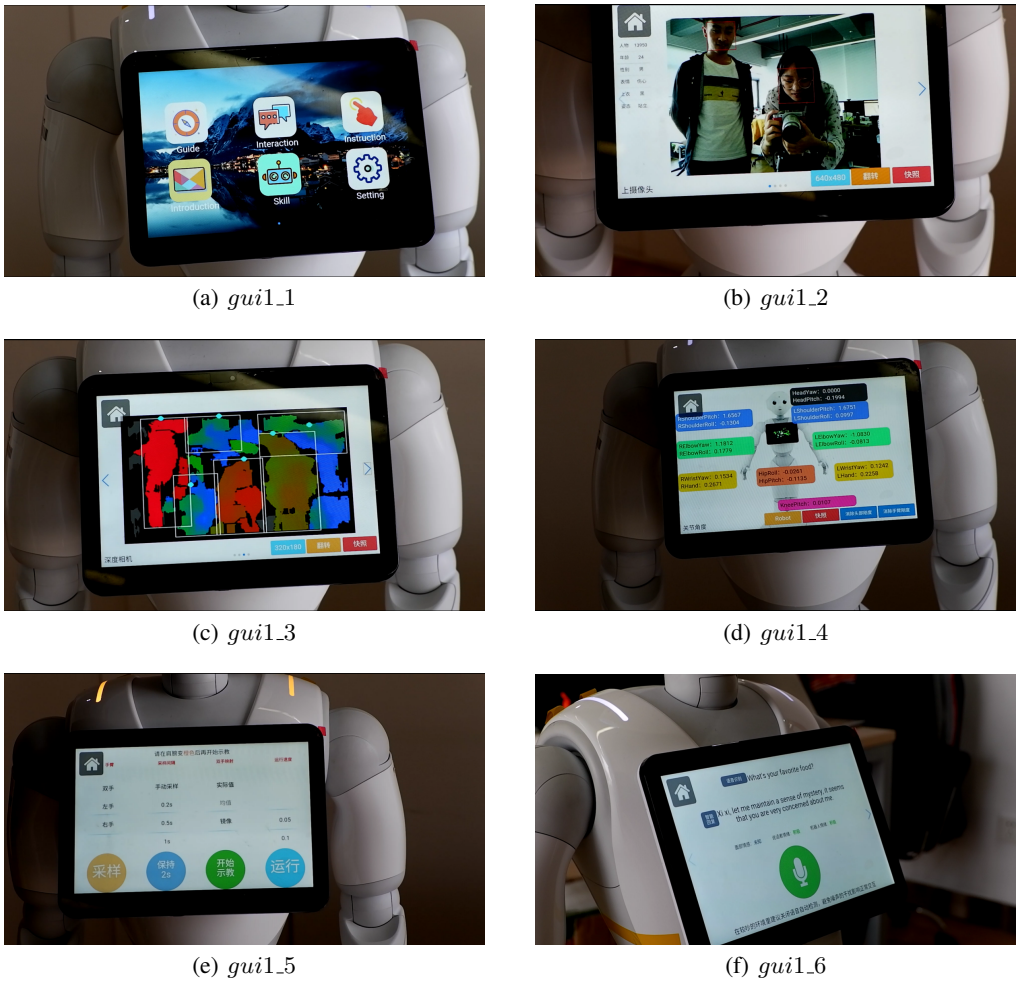
As a domestic service robot, comfortable GUI interaction is neccesary. One of the advantage of Pepper is that it has a laptop on its breast, which makes GUI possible. We designed a user-friendly and cross-platform interface so even people unfamiliar to robotics can quickly operate on Pepper. All applications implemented are integrated in this GUI.

6 Acknowledgments

We thank Samy Bengio, Jeff Dean, Matthieu Devin, Geoffrey Hinton, Nal Kalchbrenner, Thang Luong, Wolfgang Macherey, Rajat Monga, Vincent Vanhoucke, Peng Xu, Wojciech Zaremba, and the Google Brain team for useful comments and discussions.

References

[1] Thomas Wisspeintner, Tijn Zant, Luca Iocchi, and Stefan Schiffer. Robocup@home: Results in benchmarking domestic service robots. volume 5949, pages 390–401, 06 2009.



(a) $gui1\_1$ (b) $gui1\_2$

(c) $gui1\_3$ (d) $gui1\_4$

(e) $gui1\_5$ (f) $gui1\_6$

Figure 2: asdjfhskjdhf

7 team

Team Name

TJArk@Home

Contact information

He Zongtao Robot and Artificial Intelligence Lab (RAIL) Tongji University Caoan Road 4800, Jiading, Shanghai, China 1930719@tongji.edu.cn

Website

Team members

He Zongtao, Xu Weihan, Zhou Xun, Liu Zhihao, Deng Xiuqi, Wang Liuyi, Wang Naijia, Du Jiayuan, Lu Liwen

Description of hardware

Pepper by Softbank Robotics Server computer connected by WiFi

Description of software

Operation System Ubuntu 18.04/16.04 LTS; NAOqi OS; Middleware: ROS $SLAMandNavigation$: $RTAB - Map, naoqi_driver$ Object Recognition Speech Synthesis Speech Recognition