

Internship Report

Sinuo Wang
a1713814

November 20, 2022



THE UNIVERSITY
of ADELAIDE



French Australian Laboratory for Humans-Autonomous Agents Teaming

Project Area: **Interactive Machine Learning, NLP and Robotics**

Project Supervisors: **Prof. Cédric Bueche, Prof. Jean-Philippe Diguët**

1 Company Introduction

IRL-CROSSING is an international research laboratory. It is one of the flagship international collaborations from CNRS (French National Scientific Research Centre). The CROSSING Lab joined a network of more than 70 IRLs (International Research Laboratory) and is one of only five international research laboratories with industry partners in the world. They were first launched on 22 February 2021 and directed by Prof. Jean-Phillip Diguët. The term 'CROSSING' in the name represents the crossover of ideas which is the heart of IRL collaborations. As a multidisciplinary lab, they value the knowledge of researchers across different disciplines and encourage interdisciplinary collaborations. As a result of the lab's success, they strive to further collaboration efforts with researchers from different backgrounds.

The IRL directly contributes to the growth of South Australian hi-tech industries such as space, oil and gas, manufacturing and mining, defense, and space industries. It takes world-class research in the emerging field of human-machine interaction, developing new ways of efficient, ethical, and human-centred collaboration with autonomous systems. Their road map covers four domains; developing state-of-the-art models to understand and anticipate human behavior, for example, using human data to augment individual and team decision-making and team management. Providing new Energy-Efficient and Human-Based AI algorithms through cross-learning between humans and machines. Creating frameworks of autonomous agents/human interaction and understanding, such as building theoretical models of interaction and human-machine dialogue. Lastly, providing new solutions to hybrid team management, such as adaptation and regulation to reach cognitive equilibrium between humans/autonomous agents.

2 The Tasks I Carried Out

The project I took part in was developing a domestic service robot utilising interactive machine learning for the international robotics competition RoboCup@Home. The robot used in the competition is Pepper, developed by SoftBank. There are five functional modules developed in Pepper, which are reasoning, perception, navigation, movement, and interaction (dialogue). My tasks were primarily focused on robot perception and dialogue.

2.1 Robot Perception - Visualization

Our pepper perception module was pre-developed with powerful functionalities, such as human pose and object detection. To integrate all the information perceived by the robot in a user-friendly manner, we decided to use the ROS visualization tool RViz. Utilizing the detection information, all the objects and persons detected can be represented as markers shown in RViz. For reconstructing and perceiving the environment in RViz with the surrounding object detection. RViz markers require three crucial pieces of information, which are the class labels, locations, and dimensions of an objects. The original perception module only provides the object location and class labels, while dimensional information is not directly given. However, the actual dimensions of the objects in the world frame can be reconstructed through the pinhole camera model using the bounding box information in the image frame ($y_{top}, y_{bottom}, x_{left}, y_{right}$) and the distance between the object and image plane (Z_{obj}). Intrinsic camera properties were also taken into the consideration, which is

formulated into a camera intrinsic matrix K [1].

$$K = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix}$$

, where fx , fy are the x-axis and y-axis focal length of the camera in pixels, respectively; cx and cy are the offsets of the principal point from the top-left corner of the image frame. All these intrinsic parameters were available from the camera specifications. Therefore, using the pinhole camera model, the x and y world dimensions can be reconstructed as:

$$scale_x = \frac{(X_{right}-cx)Z_{obj}}{fx} - \frac{(X_{left}-cx)Z_{obj}}{fx} = \frac{X_{difference}*Z_{obj}}{fx}$$

$$scale_y = \frac{(Y_{top}-cy)Z_{obj}}{fy} - \frac{(Y_{bottom}-cy)Z_{obj}}{fy} = \frac{Y_{difference}*Z_{obj}}{fy}$$

As a result, the perceived information about the surroundings is reconstructed in RViz, as shown in Fig. 1 below.

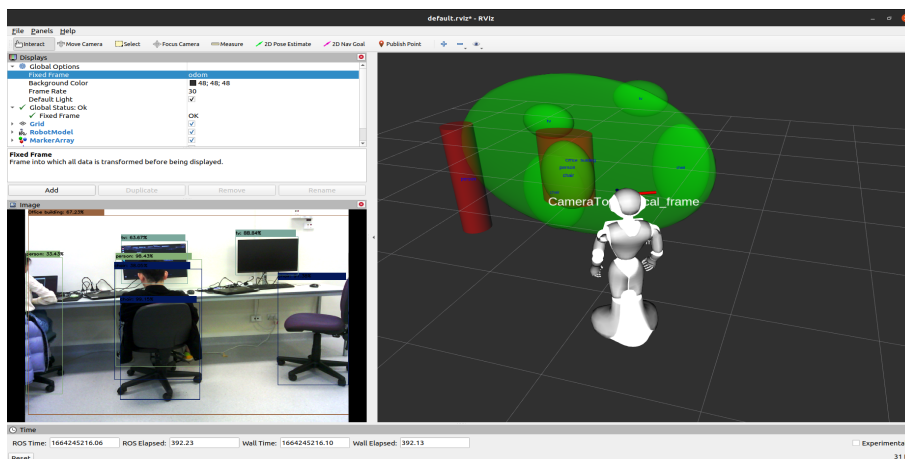


Figure 1: Robot Perceived Environment in RViz

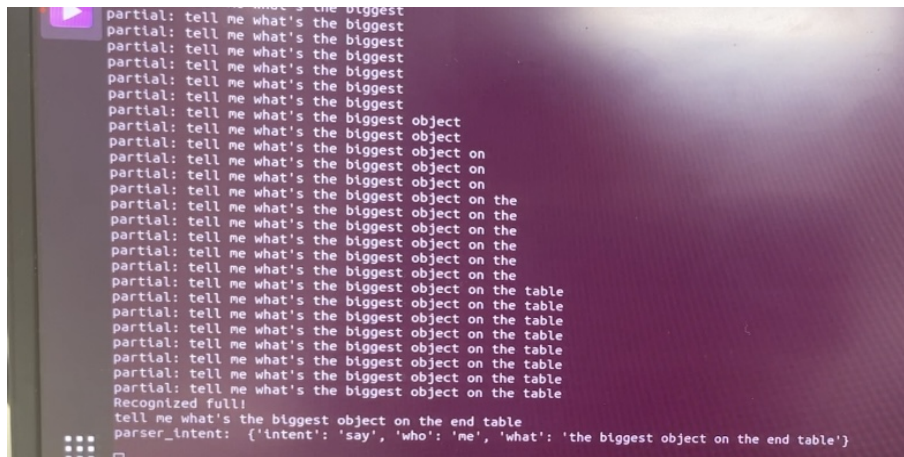
2.2 Robot Dialogue

The robot dialogue module is developed for the GPSR (General Purpose Service Robot) task in the RoboCup@Home competition. In the GPSR task, Pepper is required to process a complex user command to define a plan for accomplishing the task as requested. The Pepper dialogue module consists of two phases in the workflow: speech-to-text and text-to-intention.

2.2.1 Robot Dialogue - Speech to Text

Robustness Improvement Speech-to-text is the initial step in the process, which involves listening to user speech for identifying the converting voice commands to text. Regarding Automatic Speech Recognition (ASR) we use the lightest Vosk [2] model ("vosk-model-small-en-us-0.15") with the Kaldi recognizer [3] for noise suppression. Our original sound recording algorithm uses sound energy levels to detect whether the user is making a command and decides whether to start the recording accordingly. However, this original method showed to have a deficiency in robustness since the method is not invariant to the

background noise and natural pitch changes in human speech. To prevent cases where sound recordings stop prematurely or are delayed, we create a thread to record the sound continuously and append it to the sound data buffer. Therefore, the speech recording and our light speech-to-text inference can operate simultaneously to maintain robust but time-efficient performance. Figure 2 below shows an example of real-time ASR of the speech of command of "Tell me what's the biggest object on the table.". In the new method setting, the Vosk model inferred the text in real time and obtained the partial speech results until the entire command is recognised.



```

partial: tell me what's the biggest
partial: tell me what's the biggest
partial: tell me what's the biggest
partial: tell me what's the biggest
partial: tell me what's the biggest
partial: tell me what's the biggest
partial: tell me what's the biggest object
partial: tell me what's the biggest object
partial: tell me what's the biggest object on
partial: tell me what's the biggest object on
partial: tell me what's the biggest object on the
partial: tell me what's the biggest object on the
partial: tell me what's the biggest object on the
partial: tell me what's the biggest object on the
partial: tell me what's the biggest object on the table
partial: tell me what's the biggest object on the table
partial: tell me what's the biggest object on the table
partial: tell me what's the biggest object on the table
partial: tell me what's the biggest object on the table
partial: tell me what's the biggest object on the table
Recognized full!
tell me what's the biggest object on the end table
parser_intent: {'intent': 'say', 'who': 'me', 'what': 'the biggest object on the end table'}

```

Figure 2: Speech-to-Text Real-time Inference

Time efficiency Metric Evaluation To evaluate the time efficiency of the new method of multi-threading in recording and speech-to-text inference, we defined the metrics as $\frac{T_{speech}}{T_{inference}}$. A higher time efficiency metric indicates more efficient speech-to-text inference. I evaluated the time efficiency over 10 representative command speeches with different utterances and time lengths (T_{speech}) then the metric for all the command samples is averaged. The new method achieved 0.5190 in the efficiency metric, which outperformed the original energy level algorithm with only 0.3047 in efficiency.

2.2.2 Robot Dialogue - Text to Intention

Joint-Slot Filling Dataset Generation After obtaining the text of the user command, the Pepper robot is required to recognise the user's intention with the task-related objects or personal word entities from the text to generate a task plan accordingly. Therefore, we decided to train an NLP model to perform Joint-Slot Filling, which is a task that aims to classify utterances in intent class and then fill the related slots as arguments. State-of-the-art Joint-Slot Filling models were trained on Snips (home smart service) and ATIS (airline service), which are not helpful for GPSR tasks. Furthermore, these datasets also do not support pronoun disambiguation. For example, in the GPSR command of "Find Peter, and follow him.", the robot has to know "him" references Peter. Therefore, we decided to generate a new dataset specifically for GPSR tasks with the support of pronoun disambiguation. I used open-sourced GPSR command generator to generate 1000 user command text [4] and wrote my code to label the entities at the word token level. Figure 3 shows an example of two samples for pronoun disambiguation joint-slot filling. My future work here is to fine-tune the JointBERT model with my newly

generated dataset to provide a state-of-the-art Natural Language Understanding (NLU) solution for the GPSR tasks [5].

follow	skyler	from	the	entrance	to	the	bedroom			
B-follow	B-follow.per	O	O	B-follow.dest	O	O	O			
meet	skyler	at	the	entrance	and	accompany	him	to	the	bookcase
B-find	B-find.per	O	O	B-find.dest	O	B-take	B-take.pro	O	O	B-take.dest

Figure 3: GPSR Joint-Slot Fill with Pronoun Disambiguation Sample

2.3 Self-evaluation and Differences in Expectations

As discussed above, the three tasks I've done are robot perception visualization, robot dialogue speech-to-text, and text-to-intention. The task of perception visualization was achieved successfully and obtained the expected outcome. For robot performance testing, the team could easily use this visualization tool for quick debugging and checking for robot perception. Especially in the case of high-level GPSR tasks, the robot requires to utilize multiple functional modules, such as dialogue, perception, navigation, and manipulation. To check the correctness of the robot perception, the team does not need to read all the outputs of the labels, coordinates, and bounding boxes of all the objects detected through the terminal; instead, now we could have a quick look at the RViz and check the perception results. My next task of robot dialogue speech-to-text was accomplished successfully to my expectations. However, during development, I had difficulties with multi-threading due to my lack of background knowledge in this domain. I am very grateful to my team for helping me and providing adequate and valuable multi-threading resources so that I could achieve the task successfully. The final task of the robot dialogue text-to-intention dataset generation was also achieved. However, our ultimate goal is to fine-tune the JointBERT model to have an NLU module ready for GPSR tasks in the competition. Due to the frequent occurrence of pronoun references in the GPSR commands, pronoun ambiguity is a problem that cannot be neglected. However, pronoun disambiguation in Joint-Slot Filling is still an unsolved problem in academic literature and industry, such as the Winograd Schema Challenge. For example, in the sentence 'The trophy would not fit in the suitcase, because it was too big.', our modern models are not able to figure out whether 'it' is referred to as the 'trophy' or the 'suitcase'. In that sense, the NLU models should not only be based on linguistic statistics but also need external knowledge about the world and use reasoning to perform sentence parsing. I am doing more literature reviews about pronoun disambiguation in all the NLP tasks and devising a state-of-the-art solution with JointBERT model in my future work.

3 What Did I Learn?

3.1 knowledge Learned from the Internship

Firstly, I gained background knowledge about smart robotics, especially the crucial components and how they work together to provide powerful functionalities. For example,

LiDAR and ultrasonic sensors could be used to perceive the environment and detect obstacles. The RGB cameras provide the robot with inputs for high-level perceptions, such as object detection and human recognition. With the help of a depth camera, the robot can recover the depth information lost from the RGB cameras and support the usage of 3D PointCloud. The microphone allows the robot to listen to the users and use the audio for future NLU. The combination of the sensors enables the robot to perceive the environment with sufficient information. The integrated CPU allows the robot to run machine learning algorithms on-board so that the robot can take actions with its move base or manipulators accordingly. Secondly, I also learned to use the trick of multi-threaded programming for real-time machine learning. In most literature regarding machine learning, the focus is on accuracy improvement but less so on time efficiency. Therefore, it is good to have experience developing machine learning software for real-life use cases so that I can be aware of a comprehensive set of evaluation metrics and the corresponding improvement solutions. Lastly, I learned the NLP task of Joint-Slot Filling and the modern challenges in NLU. As discussed above, Joint-Slot Filling is a task that aims at classifying utterances in intent class and then filling related slots as arguments. For AI-based virtual assistants, such as Google Home and Amazon Alexa, Joint-Slot Filling is the task used for training their NLU for understanding user commands. The ability of AI to understand human requests is in massive demand in the modern world because these AI services mentioned above fail to understand natural language sufficiently, where ambiguities can often occur. Some modern challenges are the integration of the external knowledge [6] and common sense reasoning [7], such as the Winograd Schema Challenge [8].

3.2 Knowledge Learned from University Used in the Internship

During my involvement, my coursework knowledge provided me with a solid foundation for delivering my tasks. In the perception visualization task, I utilised the camera calibration and pinhole camera model I learned from the Computer Vision course. With this background knowledge, I could easily reconstruct the width and height of the object in the world frame from the bounding box in the image frame. The course Concepts in Artificial Intelligence and Machine Learning taught me Exploratory Data Analysis (EDA) for dataset preprocessing, so I could remove the punctuation but retain the alphabets and digits in my GPSR command dataset. Furthermore, I gained experience in ROS (Robot Operating System) when I did my Bachelor's Degree final year project. This background helped me greatly in all my tasks since everything about the robot is implemented on ROS.

4 Advice for Other Interns

4.1 Useful Background Knowledge

Firstly, proficiency in GitHub is helpful for computer science-related internships because most contemporary research is developed with version control. Another advantage of GitHub is there are many open-source state-of-the-art works. If you are doing research in one area, then you can not only read their paper but also review their code. Therefore, with the help of GitHub, you can understand the technical details of the state-of-the-art quickly. However, if you want to use their open-source code directly, please be mindful of open-source licensing. Secondly, research skill is also a plus if you are seeking an intern-

ship. In most cases of research and development, people do not have all the background knowledge they need at the beginning; instead, we need continuously learn and explore possible solutions. Therefore, research skill is essential for developers. Last but not least, coding skills are essential for computer science internships. Apart from basic coding proficiency, some background knowledge, such as algorithm design and data structures, could also help you to code smartly.

4.2 Background Knowledge Preparation

There are many great tutorials about how to use GitHub online, so you can start by learning from the tutorials. Regardless of whether the tutorial is in words or videos the most important, this is to engage. To develop research skills, you could start by solving the problems in the Kaggle competition. Kaggle is a general machine learning competition that covers the topics of computer vision, NLP, etc. Therefore, no matter what your specific research interests are, there are machine learning tasks that lie in your research area. After the competition, there are also people sharing their solutions, so people could help each other learn. Furthermore, if you have a specific research interest, then keeping yourself updated with state-of-the-art works in that area is also very helpful. For example, if you are interested in medical AI, then you could read the latest papers from MICCAI and check the state-of-the-art performance rank in each medical AI task on the website 'Paper With Code'. Lastly, for general coding skills, the courses provided by the university are helpful in developing a general computer science foundation. However, if you are seeking a method to advance your coding proficiency further, you could solve the coding problems in LeetCode. The coding problems there could not only help you practice your coding skills but also help you to prepare for technical interviews.

4.3 What to Look for in an Internship

Through the internship, you could gain practical experience and apply your knowledge in the real world. This process could help you to enhance your learned knowledge, but, most importantly, it also provides you the opportunities to realise the knowledge that you do not know. During the research and development, you are extra motivated to learn new knowledge to solve the current problem. Therefore, the internship provides you with a good environment to learn and grow. You could also get more ideas about the career path you wish to take. For example, this internship made me realise that I would like to do a research job or pursue a PhD degree in machine learning. Lastly, you could also expand your network to meet domain experts, potential employers, and colleagues to aid in your early career.

References

- [1] ANDRIES, M, BARANGE, M, BOUABDELLI, M, BUCHE, C, DIZET, A, DUHAUT, D, LEGELEUX, A, NEAU, M, PAUCHET, A, RASENDRASOA, S, et al. "Robo-Breizh 2022 Team Description Paper". In: ().
- [2] <https://github.com/alphacep/vosk>.

-
- [3] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. “The Kaldi speech recognition toolkit”. In: *IEEE 2011 workshop on automatic speech recognition and understanding*. CONF. IEEE Signal Processing Society. 2011.
 - [4] <https://github.com/kyordhel/GPSRCmdGen>.
 - [5] Chen, Q., Zhuo, Z., and Wang, W. “Bert for joint intent classification and slot filling”. In: *arXiv preprint arXiv:1902.10909* (2019).
 - [6] Xu, R., Fang, Y., Zhu, C., and Zeng, M. “Does knowledge help general nlu? an empirical study”. In: *arXiv preprint arXiv:2109.00563* (2021).
 - [7] Sap, M., Shwartz, V., Bosselut, A., Choi, Y., and Roth, D. “Commonsense reasoning for natural language processing”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. 2020, pp. 27–33.
 - [8] Camburu, O.-M., Kocijan, V., Lukasiewicz, T., and Yordanov, Y. “A surprisingly robust trick for the winograd schema challenge”. In: (2019).