From User Spoken Commands to Robot Task Plans: a Case Study in RoboCup@Home*

Mithun Kinarullathil[1,2], Pedro H. Martins[1], Carlos Azevedo[1], Oscar Lima[1], Guilherme Lawless[1], Pedro U. Lima[1], Luís Custódio[1], Rodrigo Ventura[1]

***Abstract*— In this paper, we present a functional human-robot interaction pipeline, for service robots in domestic environments, that is able to populate a Knowledge Base (KB) from sensor data with facts and goals for AI planning purposes. We use a microphone to capture human commands that are recognized and translated into text. Afterwards, we use Long Short Term Memory (LSTM) based Deep Neural Networks (DNN) for intention and arguments classification and finally an intention to knowledge component. The approach was successfully tested and used during the RoboCup@Home international competition having obtained the 4th place.**

I. INTRODUCTION

Speech is one of the natural ways of Human-Robot Interaction (HRI), and it is considered essential for service robots. Speech recognition is prone to problems based on environmental factors, such as noise [1] and the robot's own sounds. It is important for the robot to efficiently extract the information (Natural Language Understanding, NLU) [2] required to carry out the given task. Many of the off-the-shelf modules available for this task have been benchmarked [3]. In this paper, we present, i) a functional pipeline, shown in Fig. 1, which receives the human spoken commands and turns them into robot task plans, ii) the tools developed to test its performance against multiple datasets, and iii) a case study of this pipeline in RoboCup@Home competition using the MOnarCH robot.

II. AUTOMATIC SPEECH RECOGNITION

The automatic speech recognizer (ASR) provides four variable configuration parameters to reduce the adverse effects of environmental factors, such as noise and the fluctuations in the intensity of the human command. From a list of possible human commands detected by the ASR, the command with a confidence value greater than a Confidence Threshold (CT) parameter is chosen as the final output. A Human confirmation component, where the user has to approve the command detected by the ASR, was also implemented (Fig. 1) to reduce erroneous detections.

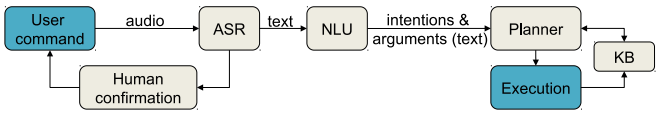

Fig. 1. Functional HRI pipeline, from user commands to robot task plans

*This work was partially supported by the FCT projects UID/CEC/20021/2013, CMUP-ERI/HCI/0051/2013, and PTDC/EEI-SII/4698/2014.

[1]Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal.

[2]mithun.kinarullathil@tecnico.ulisboa.pt

A. Development of Confidence Threshold Estimator

For estimating an optimal CT (OCT), confidence levels of multiple human commands are logged in a file. An OCT value filters the noise including erroneous detections and accepts only the spoken human command. The OCT,

$$OCT = \frac{\Sigma_i(A_i + R_i)}{N}, \quad (1)$$

was chosen as the mean of the confidence values of accepted (A_i) and rejected (R_i) commands as shown in (1), where N is the number of logged commands.

B. Development of ASR Test (ASRT)

Since manual testing can be exhausting to the user and could adversely affect the objectiveness of the test, an ASRT was developed. The commands in audio format (test dataset) were given sequentially to the ASR, and the outputs were compared to the corresponding commands in text format (test dataset). The test was also declared a failure if the ASR was not responding for a given period of time. A log and the audio feedback of the command currently being tested were presented to the user executing the test, preceding the generation of the accuracy at the end of the test.

III. INTENTION AND ARGUMENTS CLASSIFICATION

The goal of the NLU module[3] is to classify the intention and arguments posed implicitly (i.e., *proceed* and *move* implies the same intention), as shown in Fig. 2, in the human command. In Fig. 2, the output holds the necessary information required for the robot to execute the task at hand. We use two classifiers, one for intention and another for arguments classification, as presented in [2].

A. Data Collection

The training and validation datasets were synthetically generated and annotated using a developed script. To reduce the possibility of class imbalance, a two-step data pre-processing was implemented, as shown in Fig. 3. Individual class data

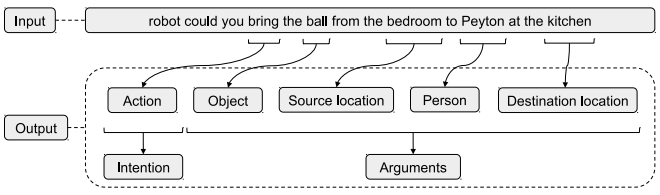

Fig. 2. An example of NLU module input and the extracted information

[3]https://github.com/socrob/mbot_natural_language_processing

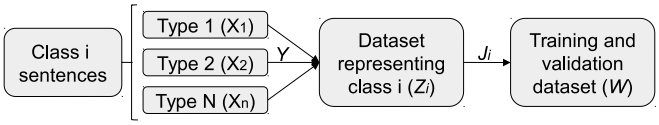

Fig. 3. Data pre-processing method to reduce class imbalance

contains n separate sets of structurally different sentences with X_i data points each. From each such set, Y,

$$Y = \frac{\Sigma_i X_i}{n}, \tag{2}$$

data points were randomly resampled to form the dataset Z_i, with a uniform representation over the specific class data. The final training and validation dataset W,

$$W = \bigcup_i J_i, \tag{3}$$

was generated by the union of sub-datasets (J_i) resampled from Z_i. The duplication of data while resampling was only allowed when it was required to generate more data from a smaller dataset.

B. DNN Training and Validation

To reduce the possibility of overfitting, the dataset was shuffled and batched before the training and validation processes. After shuffling, the dataset was split equally into ten batches, of which seven batches constitute the training dataset and the remaining three are considered as the validation dataset. The validation batch has a minimal intersection with the training batch, which is ensured by the data pre-processing method shown in Fig. 3. The underlying optimization process and the DNN architecture used for accomplishing these tasks is presented in [2]. The training, validation and the test datasets are used to evaluate the performance of the classifiers. A data visualization tool, TensorBoard[4], is used to visualize the variation in the training and the validation accuracy. The training accuracy was used to detect major problems in the training pipeline. The validation accuracy in combination with the training accuracy was used to observe the extent of overfitting.

C. Development of NLU Test Module

The test dataset is comprised of manually generated sentences and annotations, as shown in Fig. 2, with no intersection to the training and validation datasets. The test provides three types of results. First, if the intention is misclassified, then the correct and the detected intention are presented. Second, if any of the arguments are misclassified, then the correct and detected arguments are separately provided. Third, if neither the intention nor the arguments are recognized, then this message along with the correct intention and corresponding arguments are provided. The test is efficient with a processing speed of 15 sentences per second.

IV. PLANNING AND EXECUTION

The information provided by the NLU module was mapped into knowledge using the Planning Domain Definition Language (PDDL) standard as facts and goals and stored in the robot KB. If the robot KB contains unfinished goals, the planning framework [4] generates the corresponding PDDL problem, that along with the PDDL model compose the planner [5] input. The planner transforms this information into a plan, which is consequently validated and parsed using the plan validation tool [6], selecting a sequence of primitive actions to be executed. Upon action execution completion, the module updates the KB by removing finished goals and by adding/removing newly acquired facts.

[4]https://www.tensorflow.org/guide/summaries_and_tensorboard

V. RESULTS

A. ASRT Result

The ASRT was conducted using three datasets, with 20 commands each, representing General Purpose Service Robot (GPSR), Extended Endurance GPSR (EEGPSR) and Speech and Person Recognition (SPR) tasks of RoboCup@Home. SPR and GPSR tests delivered accuracies of 85% (17S, 1F, 2N) and 80% (16S, 0F, 4N) respectively, where S, F and N are acronyms for successful, failure and no response results. This result can be attributed to the good configuration parameters obtained using previously mentioned methods. EEGPSR test showed a lower accuracy of 35% (7S, 8F, 5N), which can be explained by the fact that EEGPSR test dataset has compound sentences in comparison with GPSR or SPR test datasets. The resultant configuration parameters were used for GPSR[5] and SPR[6] in RoboCup@Home 2018 by the SocRob team[7]

B. NLU Test Result

The developed tools accelerated the process of creating reliable classifiers. Two test datasets with 75 (GPSR) and 112 (EEGPSR) commands, with structural similarity but minimal intersection to the training dataset was selected initially, which yielded accuracies of 98.6% and 100% respectively. This high accuracy can be explained due to the structural similarities between the test and training datasets. To test the classifiers further, we added 25 sentences, which included sentences with synonyms of the intentions and dissimilar arguments in comparison to the training dataset. This test yielded 94.0% and 97.8% accuracies for GPSR and EEGPSR respectively.

REFERENCES

[1] A. Zermini, Q. Liu, Y. Xu, M. D. Plumbley, D. Betts and W. Wang. *Binaural and log-power spectra features with deep neural networks for speech-noise separation.* IEEE Workshop on MMSP, 2017.

[2] P. Martins, L. Custódio and R. Ventura. *A deep learning approach for understanding natural language commands for mobile service robots.* Available: arXiv:1807.03053, 2018.

[3] A. Vanzo, L. Iocchi, D. Nardi, R. Memmesheimer, D. Paulus, I. Ivanovska, and G. Kraetzschmar. *Benchmarking Speech Understanding in Service Robotics.* Proc. of the Italian Workshop on AIRO, 2017.

[4] O. Lima, R. Ventura, and I. Awaad. *Integrating classical planning and real robots in industrial and service robotics domains.* In Workshop on PlanRob, International Conference on ICAPS, 2018.

[5] M. Katz and J. Hoffmann. *Mercury planner: Pushing the limits of partial delete relaxation.* Proc. of IPC, 2014.

[6] R. Howey, D. Long and M. Fox. *VAL: automatic plan validation, continuous effects and mixed initiative planning using PDDL.* IEEE International Conference on Tools with AI, 2004.

[5]https://www.youtube.com/watch?v=Hf9bvckdTRQ

[6]https://www.youtube.com/watch?v=t5Q2ejBgPDg

[7]http://socrob.isr.tecnico.ulisboa.pt