Incrementally Learning Semantic Attributes through Dialogue Interaction

Robotics Track

Andrea Vanzo
Sapienza University of Rome
Rome, Italy
vanzo@dis.uniroma1.it

Jose L. Part
Edinburgh Centre for Robotics
Edinburgh, Scotland, UK
jose.part@ed.ac.uk

Yanchao Yu
Heriot-Watt University
Edinburgh, Scotland, UK
yy147@hw.ac.uk

Daniele Nardi
Sapienza University of Rome
Rome, Italy
nardi@dis.uniroma1.it

Oliver Lemon
Heriot-Watt University
Edinburgh, Scotland, UK
o.lemon@hw.ac.uk

ABSTRACT

Enabling a robot to properly interact with users plays a key role in the effective deployment of robotic platforms in domestic environments. Robots must be able to rely on interaction to improve their behaviour and adaptively understand their operational world. Semantic mapping is the task of building a representation of the environment, that can be enhanced through interaction with the user. In this task, a proper and effective acquisition of semantic attributes of targeted entities is essential for the task accomplishment itself. In this paper, we focus on the problem of learning dialogue policies to support semantic attribute acquisition, so that the effort required by humans in providing knowledge to the robot through dialogue is minimized. To this end, we design our Dialogue Manager as a multi-objective Markov Decision Process, solving the optimisation problem through Reinforcement Learning. The Dialogue Manager interfaces with an online incremental visual classifier, based on a Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN). Experiments in a simulated scenario show the effectiveness of the proposed solution, suggesting that perceptual information can be properly exploited to reduce human tutoring cost. Moreover, a dialogue policy trained on a small amount of data generalises well to larger datasets, and so the proposed online scheme, as well as the real-time nature of the processing, are suited for an extensive deployment in real scenarios. To this end, this paper provides a demonstration of the complete system on a real robot.

KEYWORDS

Dialogue Management; Reinforcement Learning; Interactive Learning; Incremental Learning

ACM Reference Format:
Andrea Vanzo, Jose L. Part, Yanchao Yu, Daniele Nardi, and Oliver Lemon. 2018. Incrementally Learning Semantic Attributes through Dialogue Interaction. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

One of the main steps towards the deployment of robotic platforms in real scenarios concerns their capability to reference objects and locations within the operational environment. Even though research on visual perception is pushing forward the performance of such systems, they still cannot be considered reliable enough to be used without human validation. Moreover, a purely visual perception system is often not able to provide a complete semantic description of the entities populating the environment and its output is often limited to feature representation about the world. In addition, in real deployments, a robot may need to learn idiosyncratic language used by different individuals, so that word meanings may need to be learnt and adapted through interaction. Semantic mapping is a process that aims at building a representation of the robot's world that integrates perceptual information, i.e., derived from the robot's sensors, with a semantic description of the world [16]. Hence, in semantic mapping, a major role is played by the process of semantic attribute acquisition. In recent years, several orthogonal approaches to semantic mapping have been proposed. A semantic map can be built by relying on hand-crafted ontologies and using traditional AI reasoning techniques, unable to catch uncertainty inherently connected with semantic information coming from robot sensory systems [3, 17]. The resulting map will be a static representation of the expert's perception of the world, preventing an effective adaptation to the end-user.

Figure 1: The robot used in the real scenario demonstration

Other techniques [2, 14, 25] explore semantic mapping as a process where the purely automatic interpretation of perceptual outcomes is exploited to semantically enrich a geometric map. In this setting, no human effort is required and the process is performed completely autonomously. On the other hand, a detailed structure of the semantic properties is hard to acquire, some useful semantic information could be lost, and information cannot be gained through interaction.

Few approaches consider the human as part of the loop, by exploiting interactions in a human-robot collaboration setting [5, 19]. In this case, the user is an instructor (or tutor), that helps the robot (or learner) to acquire the required knowledge about the environment. These approaches are affected by tutoring costs, as the user becomes a main source of knowledge for the robot, and their annoyance should be minimised. In fact, such online incremental learning of the targeted semantic attributes can be tedious for the tutor, whenever the robot does not exploit the acquired information to improve the interaction experience and minimise the tutoring cost. Moreover, at some point, we may want the robot to autonomously acquire new knowledge, and become more and more independent of the human, as the learning process proceeds.

In this paper we propose an interactive system for the acquisition of semantic properties of objects along with their synthetic visual representation. In the proposed approach, the interaction becomes more and more efficient over time, as the tutoring effort is quickly minimised. The dialogue policy is based on a multi-objective Markov Decision Process (MDP), while the optimisation problem is solved through Reinforcement Learning (RL). To this end, in line with [26], we argue that such cost-effective teaching can be obtained by exploiting the increasing reliability of the visual classifiers that are learned incrementally. The latter is realised by following the architecture proposed in [18], enabling an incremental online learning of object classes. The process is performed by combining a pre-trained *Convolutional Neural Network* (CNN) with a *Self-Organizing Incremental Neural Network* (SOINN). Images acquired by the robot's depth camera are preprocessed and fed to the CNN, which maps them onto a lower-dimensional feature space. With every new input feature vector, the SOINN is able to adapt in order to reflect the underlying topology of the data distribution. The whole process can be performed in real time. Hence, we hypothesise that, when the perceptual information is properly exploited, the Dialogue Manager (DM) is able to minimise the tutoring cost, resulting in a less tedious interactive mapping process.

We tested our hypothesis through several empirical investigations, whose outcomes show that the resulting adaptive dialogue strategy is able to find an optimal trade-off between the classifier accuracy and the tutoring cost. Results are encouraging for the deployment of the system in real scenarios: the experiments showed that the policies can be successfully trained on a small dataset and yet, generalise well to perform properly on larger ones.

The remainder of the paper is organised as follows. In Section 2, we survey related work on approaches for visual attribute learning. Section 3 provides a description of the proposed modular system, focusing on the visual classifier and the dialogue strategy acquisition process. The experimental setting to validate our hypothesis is presented in Section 4. In Section 5 the results are analysed, while Section 7 draws some conclusions and discusses future work.

2 RELATED WORK

In recent years, several systems have aimed at mapping the operational environment of a robot with semantic attachments. According to the definition provided in [16], the resulting "semantic map" is a representation of the environment that couples the spatial structure with semantic information concerning locations and objects therein. In this respect, such a process is often carried out by associating symbols to physical elements of the environment [7].

Several works treat the problem as a fully automated process. In [2] the authors focussed on the recognition of rooms by extracting "valuable" attributes. In [1, 4, 6, 13] topological maps are built upon the metric ones, enabling the robot to perform an "aware" navigation of the environment. With the recent advances in object recognition and categorisation, several approaches exploiting visual features have been proposed [14, 25].

A few approaches rely on the presence of a human in the learning loop, acting as a tutor who instructs the robot to learn the environment. In fact, fully automated semantic mapping systems are error-prone and do not provide the wide-range knowledge that can be acquired by interacting with a user through speech. For example, in [15] the authors rely on a multivariate probabilistic model to associate semantic labels to spatial regions and on the support of the user in selecting the correct one. Conversely, in [11] clarification dialogues are used to support the mapping process. Such an approach is further extended in [27] to create conceptual representations of indoor environments which are used in human-robot dialogue. More recently, in [5] a human-augmented semantic mapping system is presented. The authors focus on an online setting, where the semantics of objects is acquired incrementally through long-term interactions with the user. The dialogue policy is implemented beforehand through Petri Net Plans (PNPs) and the robot is not enabled to infer semantic properties of new objects from the acquired knowledge. In [23], the authors propose an approach for the opportunistic acquisition of objects descriptors (or attributes) relying on the users' feedback. Though the task is similar, our system focuses mainly on the minimisation of the tutoring cost during the teaching activity.

This paper makes several contributions. First, in contrast with most of the previous research, the Dialogue Manager driving the semantic attribute learning is entirely data-driven. This feature is essential to enable the deployment and optimization of a robotic platform in heterogeneous environments, interacting with different users speaking different languages. In addition, our approach exploits an incremental object classifier to acquire visual information about the objects. Such knowledge is then used to automatically recognise new objects, supporting a quick acquisition of the semantic map. The dialogue interactions benefit from an analysis of the visual classifier reliability, as this information is exploited to determine whether to ask the human tutor a clarification question or not. Finally, the acquisition of the policies can be performed on a very small set of examples, while still showing good performance when tested on larger datasets. This feature enables the deployment of our approach in a long-term mapping scenario. Moreover, the policies may be further improved while the system is operating and, hence, adapted to the specific user.

Table 1: Dialogue Examples from the synthetic Dialogue Collection: *(a) the user takes the initiative (b) the learner takes the initiative.*

(a) Tutor Initiative	(b) Learner Initiative
T(utor): what is this object?	L: a shampoo bottle, am I right?
L(earner): I have no idea.	T: no, it is not.
T: a shampoo bottle.	L: so what is it?
L: okay, shampoo bottle.	T: an apple.
T: good job.	L: okay, got it.

3 AN INCREMENTAL APPROACH TO SEMANTIC ATTRIBUTE LEARNING

Semantic mapping is a complex task that involves several subproblems, such as representation of the semantic properties, route planning, interaction management, and sensing. In this work, we focus on two of them: (i) the management of the dialogue for the acquisition of semantic properties, and (ii) the memorization of synthetic representations of the object that are used to compose the semantic map. To this end, we describe hereafter the proposed interactive multi-modal system in support of learning semantic maps (e.g., visual classes) through natural conversational interaction with human tutors. Table 1 shows examples of possible interactions where tutor and learner interactively exchange information about the category of a particular visual object.

3.1 Overall System Architecture

In this section, we introduce the proposed system architecture (see Fig. 2), which loosely follows that of [26] and employs two essential modules: a *Vision Module* and a *Dialogue Module*.

Vision Module. The vision module is based on the system proposed in [18], which accomplishes incremental online learning by combining an adapted version of a Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN) [28] and a pre-trained Convolutional Neural Network (CNN) based on the architecture proposed in [10]. Through this combination, we can leverage the great representational power of deep CNNs while retaining the ability to adapt to novel input incrementally provided by the LB-SOINN. The latter is a fundamental requirement for robots operating in real and dynamic environments, as it is impossible to anticipate all the possible situations the robot may encounter during its operation.

The system consists of two channels, processing RGB and depth images respectively. Both channels resize and rescale the input images to the format expected by the CNN. The depth channel further processes the depth image to produce a colourized surface normals image. Once the images have been preprocessed, they are fed into identical pre-trained CNNs that output the corresponding feature vectors. These feature vectors are then combined by computing their average: the result is used to adapt and grow the LB-SOINN. Effectively, this module allows us to ground noun words such as "apple" and "shampoo", which are used as parameters of the Dialogue Acts in the dialogue module, onto their visual representations.

Dialogue Module. This module relies on a classical architecture for dialogue systems, composed of Dialogue Management (DM) and Natural Language Understanding (NLU), as well as Generation (NLG) components. These components interact via Dialogue Act representations [21] (e.g., $inform(obj = apple)$, $ask(object)$). The Natural Language Understanding component processes human tutor utterances by extracting a sequence of key patterns, slots and values, and then transforming them into dialogue-act representations, following a list of hand-crafted rules. The NLG component makes use of a template-based approach that chooses a suitable learner utterance for a specific dialogue act, according to the statistical distribution of utterance templates from synthetic dialogue examples. Finally, the DM component is implemented with an optimised learning policy using Reinforcement Learning (see Sect 3.3). This optimised policy is trained to: (i) process Natural Language conversations with human partners, and (ii) achieve a better trade-off between classification performance and the cost incurred by the tutor ,e.g., time and effort, in an interactive learning process.

3.2 Visual Object Classification

As mentioned in Section 3.1, one of the main components of our vision module is the LB-SOINN. This method is based on the Self-Organizing Map [9] and is able to learn the underlying topology of the data distribution, without the need to specify the number of classes in advance. Each node in the network has an associated weight which lives in the feature space. Every time a new image is input to the vision module, the LB-SOINN algorithm assesses whether a new node has to be added to the network, based on the feature vector similarity to all the other nodes' associated weights. If no node is added, then the closest node and its neighbours' weights are updated, and the two closest nodes are joined by an edge. In this manner, the structure of the network evolves to reflect how the data is distributed in the feature space.

In this work, we focus on object classification as opposed to image classification. To this end, we consider the contributions of all the images corresponding to the same object in order to produce a classification result. This procedure is consistent with a real scenario, where a robot may look at the same object from different views and infer what it is, based on the consensus achieved from the results for every individual view.

In order to classify each image, we compute a confidence score as the average inverse distance between its feature vector and the weights of every node belonging to a given class, i.e., the closest the feature vector is to the nodes corresponding to the given class, the bigger the score that class will receive. We repeat this for every class in the network and then normalise the resulting confidence scores, such that they resemble probabilities, i.e., their sum is equal to 1. Hence, the class that receives the highest normalised score is chosen as the classification result for that image. The normalised confidence score is computed as follows:

$$conf_i = \frac{1}{\sigma}\frac{1}{n_i}\sum_{j=1}^{n_i}\frac{1}{D(p, q_j)} \tag{1}$$

where n_i is the number of nodes for the i^{th} class, σ is the normalizing factor and $D(p, q_j)$ is the distance between the feature vector p and the weight q_j corresponding to the j^{th} node of the i^{th} class.

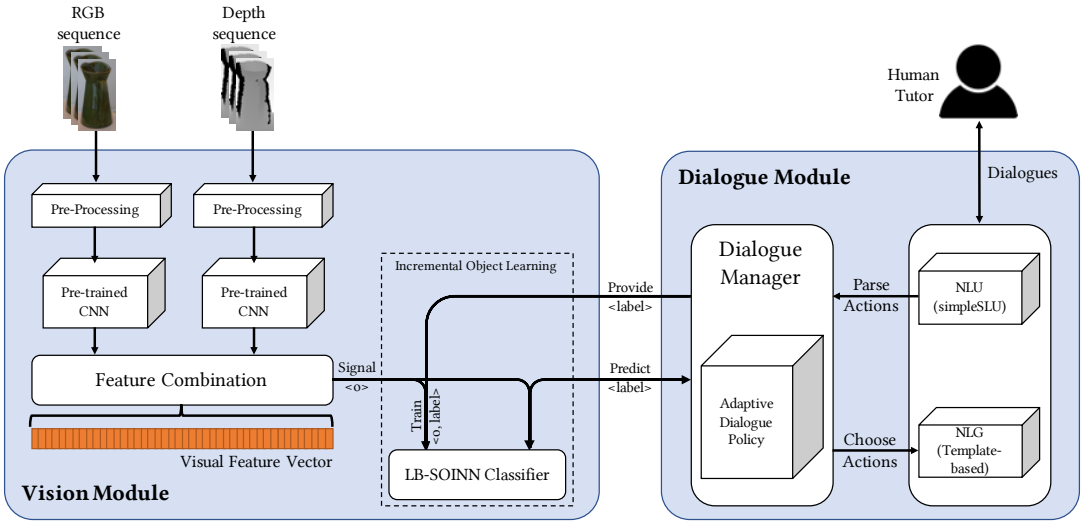

Figure 2: Overview of system architecture for semantic attributes learning

The normalizing factor is simply:

$$\sigma = \sum_{i=1}^{N} \left(\frac{1}{n_i} \sum_{j=1}^{n_i} \frac{1}{D(p, q_j)} \right) \tag{2}$$

where N is the number of classes already learned. For the computation of $D(p, q_j)$, we follow the procedure proposed by [28], where they use a combination of Euclidean and cosine distances affected by a weight that is a function of the dimensionality of the feature vector, i.e., for low-dimensional features, the Euclidean distance will be dominant whereas for high-dimensional features, the cosine distance will be dominant:

$$\begin{aligned} D(p, q_j) = \frac{1}{\eta^d} \frac{EU_{pq_j} - EU_{min}}{1 + EU_{max} - EU_{min}} + \\ + \left(1 - \frac{1}{\eta^d}\right) \frac{CO_{pq_j} - CO_{min}}{1 + CO_{max} - CO_{min}} \end{aligned} \tag{3}$$

where d is the dimension of the feature vector, $\eta = 1.001$ is a pre-defined parameter, EU_{pq_j} is the Euclidean distance between p and q_j, EU_{max} and EU_{min} are the maximum and minimum Euclidean distances between any two nodes in the network respectively, and CO_{pq_j}, CO_{max} and CO_{min} are the equivalent quantities corresponding to the cosine distance measure.

Finally, we adopt a voting schema over all the images of the object, normalising the number of images classified for any given class over the total number of images. The final result of an object classification is defined as the class that obtains the highest consensus among all the images, i.e., highest "probability".

3.3 An Adaptive Dialogue Strategy for Interactive Mapping Tasks

A comprehensive teachable system should learn as autonomously as possible, rather than involving the human tutor too frequently Skočaj et al. [20]. Accordingly, as pointed out in [26], an intelligent agent should provide the capability of finding an optimised trade-off between the goal achievement and the tutoring cost in a particular task. In other words, given the visual mapping task, the agent should be able to learn the visual scene accurately, and with little effort from human instructors. In order to optimise the trade-off, the interactive mapping problem can be formulated into two sub-tasks, i.e., *when* and *how* to learn the mappings, which are trained using Reinforcement Learning with a multi-objective Markov Decision Process (MDP), consisting of two sub-MDPs. The robot behaviour is characterised by the following sequence of steps: (i) a visual instance is shown to the agent/learner; (ii) based on the outcome of the instance classification, i.e., a confidence score for each category acquired so far, the agent/learner determines *when* and *how* to ask questions; (iii) the dialogue continues with a response from the user. The output of the first MDP (*adaptive threshold*) will be applied to determine the initial state of the second MDP (*dialogue control*), see more details as below:

3.3.1 When to Learn. In the first MDP, the policy is required to learn when the learner needs to acquire useful information from human tutors, where a form of active learning is taking place: the agent learns to ask questions about particular objects only whenever it is uncertain about its own predictions. Following work from [26], we adapt a positive confidence threshold, which determines when the learner can trust its visual predictions. This threshold plays an essential role in achieving an optimal trade-off between the classification performance and the tutoring cost, since the learner's behaviour (e.g., whether to seek feedback from the tutor or not) is dependent on this threshold.

Here, we learn an adaptive strategy that aims at maximising the overall performance by properly adjusting the confidence threshold in the range from 0.9 to 1[1].

[1]Here, we set the confidence threshold within a small range, because the output of the classifier never dropped below 0.9 through our experiment.

State Space. The adaptive-threshold MDP operates on a 2D state space, consisting of *curThreshold* and *levelRel*. *curThreshold* represents the positive threshold the agent is currently applying. *levelRel* is applied to locally measure the reliability of the visual classifier on a single learning step. To this end, the total number of instances (objects) is clustered into bins, with each bin containing n_B instances and representing a single learning step. Then, the Local Accuracy (Acc_{loc}, see Section 4.1) of classifiers is scaled between -1 to 1 ($Acc_{loc}^{[-1,1]}$) and *levelRel* is discretised into three levels as below:

$$levelRel = \begin{cases} 1, & \text{if } Acc_{loc}^{[-1,1]} > 0 \\ 0, & \text{else if } Acc_{loc}^{[-1,1]} = 0 \\ -1, & \text{otherwise} \end{cases} \quad (4)$$

Action Selection. Based on the previous performance of the classifier on a single learning step, the model updates its state space by either increasing/decreasing the current confidence threshold by 0.02, or keeping it at the same value.

Reward Function. Here, we introduce a local reward function R_{loc} for the learning task that is proportional to the local accuracy of the visual classifier, computed over each bin of n_B instances (more details about the local accuracy in Section 4.1). We rescale the local accuracy to be in the range $[-1, 1]$ to evaluate the effectiveness of the selected action. The system will reward the action if the rescaled accuracy is greater than 0, otherwise, the action is penalised.

Each training episode terminates when the agent passes through all instances in the visual dataset.

3.3.2 How to Learn Using Dialogue. The second MDP aims at acquiring useful information through interaction with human partners. For example, if the learner has a low confidence on its predictions, i.e., the confidence score is lower than 0.5, it may ask Wh-questions to acquire correct labels directly from the tutor (e.g., "*what is this object?*"). Otherwise, the learner should be able to make a guess about the label by asking a Yes-No-question (e.g., "*is this an apple?*"). In addition, the learner is also required to produce coherent conversations with a human partner, i.e., understand particular dialogue intents from humans and properly produce the next responses. In order to achieve these goals, the Reinforcement Learning process and the corresponding MDP have been configured as follows:

State Space. The dialogue policy initialises a 3D state space, defined by *cStatus*, *preDAts* and *preContext*. *cStatus* is applied to represent the current status of predictions about a particular object. The status level is determined by the confidence score ($conf$) and the positive threshold ($curThreshold$) described above (see Equation 5); *preDAts* represents the actions the tutor performed in the previous dialogue turn; *preContext* represents whether a visual category was mentioned in the dialogue history and what category it is. In our paper, as we only take into account the class name of the visual object, *preContext* may only contain one out of two values, i.e., unmentioned (U) and object class (C).

$$cStatus = \begin{cases} 2, & \text{if } conf > curThreshold \\ 1, & \text{else if } 0.5 \geq conf \geq curThreshold \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Action Selection. The actions are chosen based on the statistics of task-oriented dialogue actions occurring in a set of hand-crafted dialogue examples (see Table 1), including Wh-questions, Polar-questions, DoNotKnow, Acknowledgement, as well as Listening.

Reward signal. The reward signal is defined by a global function R_{glob}, which takes into account the cumulative cost by the tutor (*Cost*, defined in Section 4.1) in a single conversation and penalties (*Penal*) for inappropriate actions performed by the learner (e.g., if the learner does not answer a question).

$$R_{glob} = 10 - Cost - Penal \quad (6)$$

Each single dialogue represents an episode and is terminated when the class name is either taught by a human tutor or inferred through a sufficiently high confidence score. The SARSA algorithm [22] is used to learn the adaptive dialogue policy, with a greedy exploration rate of 0.1 and a discount factor of 1.

4 EXPERIMENTAL SETUP

The experimental setup aims at simulating a semantic mapping task, where the robot navigates throughout the environment to acquire semantic properties of the objects populating its world. Notice that while the problems of planning and navigation are out of the scope of this paper, we focus on the *category* (or label) of objects (e.g., `apple`, `calculator`, ...). However, the approach can be easily extended for the acquisition of other properties, such as *colour* and *shape*.

Figure 3 shows the GUI used to visualise the simulated environment. The robot navigates the assigned area, seeking for unknown objects (*red* squares in the grid). Once an item is reached, the visual classifier is fed with with images corresponding to the current instance (e.g., on the bottom right box in Fig. 3). The confidence score provided by the visual classifier is then used for deciding whether to assign the predicted label (without an interaction with the user), or to ask a Polar/Wh-question, according to the current threshold level. At the end of the interaction, the object is finally labelled with the category provided by the user[2], and the corresponding images are used to train the visual classifier. It is worth noting that the classifier is updated only whenever the label is provided by the user. That is, when the agent/learner trusts the classifier, we assume that that specific instance is already well represented in the model. Such a conservative approach aims at avoiding possible noise introduced into the net when unnecessary images are learnt.

4.1 Evaluation Metrics

To evaluate the trained learning/dialogue strategy, we make use of a measure metric based on the PARADISE evaluation framework by [24] for task-oriented dialogue systems – the overall performance of a robot takes into account both the task success (classification accuracy) and cost (the tutoring effort within dialogues). The system in the experiment is required to achieve and retain a better trade-off between the accuracy and the cost through an interactive learning period. More details about these metrics are described below.

[2]We do not deal with lexical variation; categories are identified through a vocabulary.

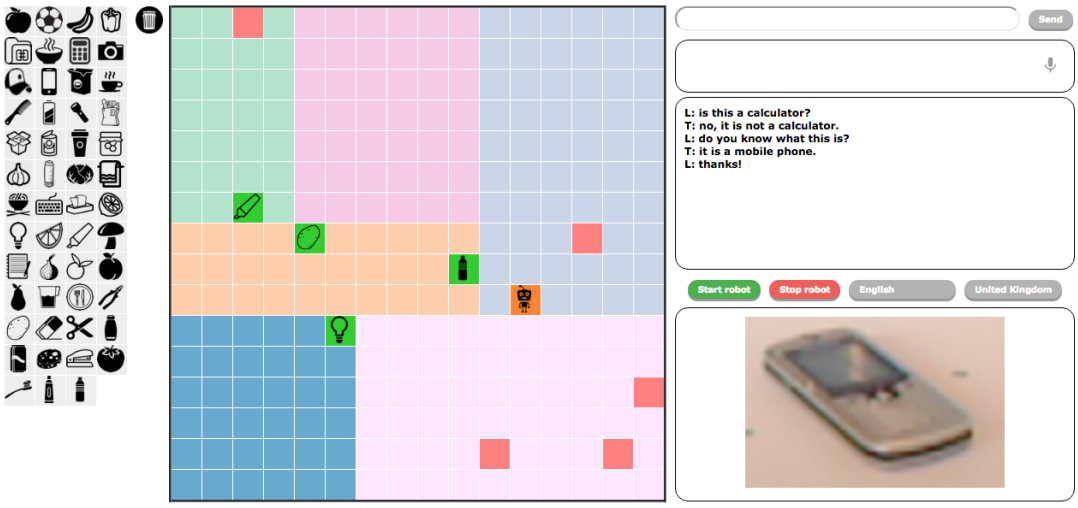

Figure 3: The simulated environment for interactive semantic attributes acquisition. The *left* block shows the labels available within the dataset; the grid map in the *centre* emulates the environment in which the robot is moving, where green cells refer to correctly recognized objects, red cells are the objects that have not already been discovered, while the orange cell is the target object; on the *right*, the dialogue flow and the images of the target object are shown

Local Accuracy. In contrast with [26], instead of using a distinct test set, we measure the learning performance of the agent using visual instances which may have been seen in previous learning steps. Hence, the system is able to self-test on objects that it has seen before, as its learning progresses. To this respect, the *Local Accuracy* (Acc_{loc}) of the i-th bin is computed at the end of the bin using the initial predictions obtained for each instance during the processing of the bin.

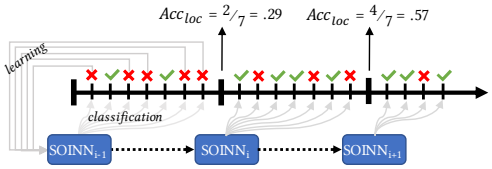

Figure 4: Local Accuracy evaluation

Operationally, as sketched in Figure 4, for each instance in the dataset, we get the prediction from the visual classifier and if the prediction is correct, the True Positives (*TP*) are increased by 1, otherwise the instance is learnt. When the n_B instances of the bin have been processed, we evaluate the *Local Accuracy* as follows:

$$Acc_{loc} = \frac{TP}{n_B} \tag{7}$$

and reset the *TP*. Hence, the evaluation score obtained after each bin is not biased by the training data. In fact, the prediction of each object is made with the model acquired so far and the object is learnt only if the prediction is wrong.

Cumulative Tutoring Cost. Intuitively, cost computation is based on any user or agent dialogue turns. Skocaj et al. [20] pointed out that a comprehensive system must be able to learn as autonomously as possible, rather than involving the tutor too frequently.

In this paper, we take into account a cumulative tutoring cost (or simply *Cost*), which is only applied to reflect the effort needed by a human **tutor** in interacting with the system/robot. In the literature, a wide range of cost measures have been proposed and exploited. Given the learning task, there are four possible costs[3] that the tutor might incur, as defined below:

- C_{inf} (*Inform*) refers to the cost of the tutor providing information on the name of the specific visual instance (e.g., "*this is a shampoo bottle*"); it may be either 5 or 0, depending on whether the dialogue act is present or not within the sentence;
- C_{ack} (*Acknowledgement*) is the cost for a simple confirmation (like "*yes*", "*right*"); it is set to be 0.5;
- C_{reject} (*Rejection*) is the cost for a simple rejection (such as "*no*", "*it is wrong*"); it is set to be 0.5;
- C_{crt} (*Correction*) is the cost of correction of a statement/polar question (e.g., "*no, it is an apple*"); it is set to be either 5 or 0.

In this paper, the cumulative cost is considered as sums of these action-costs across all dialogues (i.e., a single dialogue is considered as talking about a single visual instance).

$$Cost = \sum_{i=0} C^i_{inf} + \sum_{j=0} C^j_{ack} + \sum_{k=0} C^k_{reject} + \sum_{l=0} C^l_{crt} \tag{8}$$

4.2 Visual Object Dataset

To evaluate our proposed system, we used the Washington RGB-D Object Dataset [12]. This dataset was acquired using a Kinect-like sensor and consists of 300 household objects organized into 51

[3]We also tested the cost function with other values (see details in [? ? ?]). Although somewhat arbitrary, this does not affect the overall performance, as long as there is a significant difference between "inform" and "acknowledge/rejection".

categories. For each object, video sequences of full 360° rotations at three different heights of the sensor were acquired. In addition, the dataset is provided with cropped versions of the objects and binary masks which aid in pre-processing the images. The particular nature of this dataset, i.e., the sensor used and the acquisition setup, allows us to simulate a real interactive scenario, where the robot is able to obtain different views of the same object in a sequential manner. In this work, we assume that modules for detecting and segmenting the objects would be available in a full system implemented on a robotic platform and thus, fall outside the scope of our methodology.

In order to make our analysis even more realistic, we further reduced the size of the dataset by considering only 120 random images per object and we randomly shuffled them on a per object basis. This resembles a less organized way of collecting the data, as opposed to the more structured protocol used in [12]. Then, we trained on a random subset that accounted for 50% of the images and tested on a random subset that accounted for 25% of the images. This allowed us to speed up the learning process as well as to increase the degree of overlap between train and test subsets. The latter aims to account for the fact that in a particular environment, the robot may run into the same objects several times and even though it may see them from different perspectives, chances are that it will get similar views some of the times.

4.3 User Simulation for the Learning Task

In order to train and evaluate the dialogue agent, we built a user simulation that resembles human behaviours on the task of teaching visual objects using a generic n-gram framework. The simulated tutor is trained on a collection of synthetic dialogues (see dialogue examples in Table 1), where the user's action (e.g., INFORM, NEGOTIATION, REJECTS, ...) is predicted probabilistically. This simulation framework takes as input the sequence of N most recent words in the dialogue, as well as some optional additional conditions, and then outputs the next user response on multiple levels as required (e.g., full utterance, a sequence of dialogue actions, or even a sequence of single word outputs for incremental dialogue behaviour). In this paper, we created an action-based user model that predicts the next user response in a sequence of dialogue actions. The simulator then produces a full utterance by following the statistics of utterance templates for each predicted action.

5 RESULTS

We run several empirical evaluations aiming to determine the effectiveness of the adaptive-threshold MDP and the applicability of the approach in real scenarios.

The policies have been trained for 5000 episodes on a small amount of data within this experiment, only including 48 instances, distributed over 10 classes randomly drawn from the Washington RGB-D Object Dataset. In order to clearly visualise the trend of overall performance of the learning agent, we group these instances in bins with $n_B = 8$ objects each. Table 2 provides example interactions between the learned RL agent and the simulated user, showing how the learner, in order to minimise the cost, favours taking the initiative. Afterwards, we tested the policies as follows. In order to prove the effectiveness of the policies over unseen objects, we tested them on a dataset of 25 classes (143 instances), where the overlap with the training set is minimal, repeating the experiment for 10 folds. The size of the bins was $n_B = 9$. In our scenario the robot keeps navigating the environment, so after a while it may reach an object that it has already seen before. To simulate this, we replicated the number of instances by 2, randomly shuffling the dataset, both when training and testing the policies. For example, the instance `apple_2` will be processed twice.

Table 2: Example conversations between the RL-based Learning Agent (L) and the Simulated User (T): *(a) Learner with low confidence (b) Learner with higher confidence.*

Dialogue Example (a)	Dialogue Example (b)
L: what is this object called? T: an apple L: okay, apple T: good job.	L: this is a shampoo, right? T: no, it is not shampoo, it is a stapler. L: okay, got it.

In Figure 5 we report the plots obtained from the experiments. Results are provided in terms of *Local Accuracy* (left) and *Cumulative Tutoring Cost* (right). In our analysis, we compared three different approaches for adjusting the confidence threshold.

In the first setting, we used a *Fixed threshold* (*FT*) set to 1. This is the baseline, where the robot keeps asking questions, as the classifier outcomes are always less than (or equal) to 1. The second setting relies on a *Rule-based adaptive threshold* policy (*RT*) to adjust the threshold. This hand-crafted policy modifies the threshold as follows: whenever the ΔAcc_{loc} is positive, the threshold is decreased by 0.02; if the ΔAcc_{loc} is negative, it is increased by 0.02; otherwise it is not modified. Finally, we tested the policy acquired through the approach described so far (*RL-based adaptive threshold*, or *RLT*).

We performed an ANOVA test to evaluate the significance of the different settings for Acc_{loc} and *Cost*. The outcomes suggest that there are no significant differences on the local accuracy under the three different threshold conditions. However, this is not true for the *Cost*, where the p-value is $p < 1 \times 10^{-14}$. The ANOVA results are confirmed by a post-hoc pairwise comparison over the *Cost*, performed through t-tests. The outcomes show that the *RLT* policy has significantly less tutoring cost than the others, namely the *FT* ($p < 4 \times 10^{-14}$) and the *RT* policies ($p < 0.006$).

As expected, in the first setting the *Cost* is represented by a straight line, as the learner applies the same dialogue pattern for all the interactions. Since the user always provides a label for the given object, we would expect a better Acc_{loc} curve. Instead, it seems that this metric is affected by "noise": even though the classes are well represented within the LB-SOINN model, the learner keeps updating the network by injecting unnecessary examples. The *RT* policy seems to get acceptable results, as (i) the tutoring cost tends to decrease as more objects are processed, while (ii) the accuracy is not degraded. In addition, the *RLT* setting seems to outperform the other techniques. In fact, the *Cost* is always minimised and most importantly, starts to decrease from the very beginning of the process, i.e., the threshold is decreased as soon as the robot starts to trust the classifier. This behaviour is essential, as the benefits of the RL-based threshold will be perceived even after few interactions. At the same time, the Acc_{loc} curve seems to follow the same trend of the other settings, suggesting that the tutoring cost can be

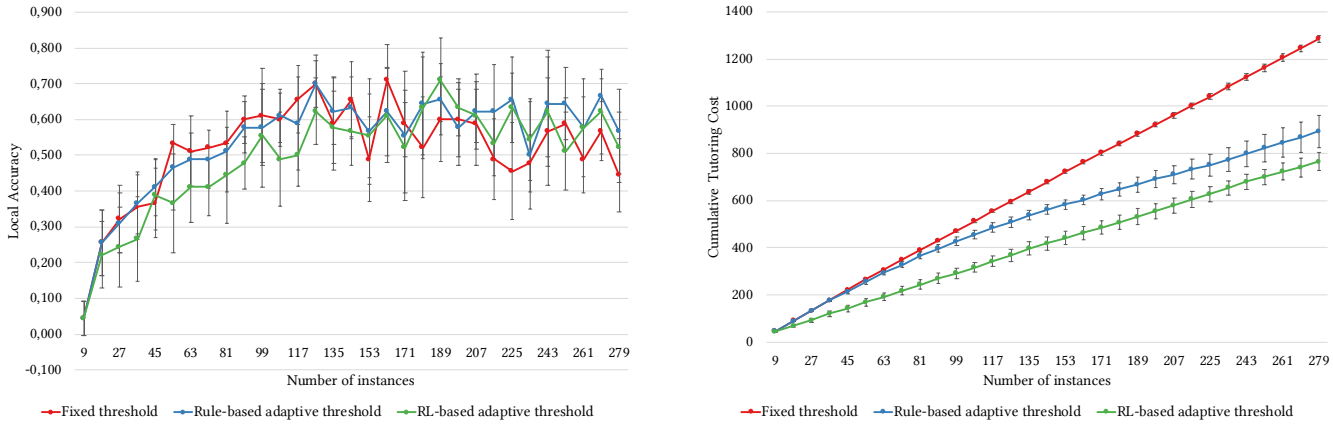

Figure 5: Results of the experimental evaluation, provided in terms of *Local Accuracy* (left) and *Cumulative Tutoring Cost* (right), along with 95% Confidence Intervals.

minimised without loss in accuracy. Nevertheless, we noticed that once a considerable number of classes is acquired, the confidence values provided by the visual classifier are lower than in the early stages, due to a higher internal uncertainty of the network. Hence, even though the prediction for an instance is correct, but with a low confidence score, the threshold does not have chance to decrease further since its lower bound is set to 90 (we chose a conservative solution for the classifier trust). As a consequence, the *Cost* stops decreasing and the corresponding curve appears as a straight line. This aspect could be further optimised in future work.

6 DEMONSTRATION ON REAL ROBOT

In order to support the effectiveness of the proposed approach, we performed a preliminary deployment of the system on a real robot[4]. The targeted platform is a modified version of the TurtleBot 2 Robot[5] (see Figure 1). While the base has not been modified, the structure on top is customized, in order to make the robot taller with respect to the off-the-shelf version. The robot is 107 cm high and features a tablet as an interface for the interactions. In fact, the ASR module has been realized through the Google Speech APIs [8], available within the Android environment, in an ad-hoc mobile application. The robot has been equipped with the Asus Xtion Pro Live RGB-D camera. Though the nature of the resulting dataset is still the same as in the simulated scenario (for each shot, RGB and depth images are taken), the presence of a textured background and the hand holding the object might interfere with the learning process (segmentation and cropping of the object are outside the scope of the paper). The video shows some interactions obtained through the policy acquired during the simulated experiment, over the dataset composed of 48 instances (10 classes). The robot was tele-operated by the user. In fact, as it was not able to autonomously detect the presence of an object in front of the camera, it was forced to capture 30 RGB-D images on command by the user, i.e., by pressing a button on the joystick controller. Then, the pipeline proceeds as in the simulated experiment. Although we did not measure the performance, the system behaves as expected, minimising the effort needed by the human in instructing the robot to acquire new objects. This further demonstration provides a preliminary evidence of the effectiveness of the proposed solution.

7 CONCLUSION

This paper focused on the problem of acquiring a dialogue policy to support interactive semantic attribute acquisition, with the goal of minimising the human users' tutoring cost. To this end, we proposed a multi-objective MDP Dialogue Manager, where the optimisation problem is solved through Reinforcement Learning and the interaction is made dependent on visual information. In fact, while one MDP is devoted to the selection of the proper Dialogue Act, the other one modifies the level of trust in visual information. The latter is provided by an online visual classifier, based on a Load-Balancing Self-Organizing Incremental Neural Network. We proved the benefits introduced by the adaptive threshold MDP through empirical investigations that confirmed our initial hypothesis. Nevertheless we consider this work as a starting point for a future line of research. First, the online schema proposed here, as well as its real-time processing, allowed for a preliminary deployment of such system in a real scenario. This will enable a thorough evaluation on a real robot interacting with real users. Second, the investigation of more accurate metrics to evaluate the reliability of the visual classifier (e.g., entropy, robustness, ...) could be beneficial for the policy acquisition. Finally, in this work we focused on the category property of an object, but a vast plethora of semantic properties could also be taken into account (e.g., *colour*, *affordances*, ...). To this end, different MDP design patterns could be explored.

ACKNOWLEDGMENTS

This research is partially supported by the EPSRC, under grant number EP/M01553X/1 (BABBLE project[6]), and by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688147 (MuMMER project[7]).

[4]https://youtu.be/jKGSuEHmDWU
[5]http://www.turtlebot.com/turtlebot2/
[6]https://sites.google.com/site/hwinteractionlab/babble
[7]http://mummer-project.eu/

REFERENCES

[1] E. Brunskill, T. Kollar, and N. Roy. 2007. Topological Mapping using Spectral Clustering and Classification. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3491–3496.

[2] P. Buschka and A. Saffiotti. 2002. A Virtual Sensor for Room Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1. 637–642.

[3] Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernandez-Madrigal, and Javier González. 2005. Multi-hierarchical semantic maps for mobile robotics. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2278–2283.

[4] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez. 2005. Multi-Hierarchical Semantic Maps for Mobile Robotics. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2278–2283.

[5] Guglielmo Gemignani, Roberto Capobianco, Emanuele Bastianelli, Domenico Daniele Bloisi, Luca Iocchi, and Daniele Nardi. 2016. Living with Robots: Interactive Environmental Knowledge Acquisition. *Robotics and Autonomous Systems* 78, Supplement C (2016), 1 – 16.

[6] Nils Goerke and Sven Braun. 2009. Building Semantic Annotated Maps by Mobile Robots. In *Proceedings of the Conference Towards Autonomous Robotic Systems*. 149–156.

[7] Joachim Hertzberg and Alessandro Saffiotti. 2008. Editorial: Using Semantic Knowledge in Robotics. *Robotics and Autonomous Systems* 56, 11 (2008), 875–877.

[8] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Signal Processing Magazine* (2012).

[9] Teuvo Kohonen. 1990. The Self-Organizing Map. *Proc. IEEE* 78 (1990), 1464–1480.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*. 1106–1114.

[11] Geert-Jan M. Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I. Christensen. 2006. Clarification Dialogues in Human-Augmented Mapping. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction (HRI '06)*. ACM, 282–289.

[12] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. 2011. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset.. In *ICRA*. IEEE, 1817–1824.

[13] O. M. Mozos and W. Burgard. 2006. Supervised Learning of Topological Maps using Semantic Information Extracted from Range Data. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2772–2777.

[14] Oscar Martinez Mozos, Hitoshi Mizutani, Ryo Kurazume, and Tsutomu Hasegawa. 2012. Categorization of Indoor Places Using the Kinect Sensor. *Sensors* 12, 5 (May 2012), 6695–6711.

[15] C. Nieto-Granda, J. G. Rogers, A. J. B. Trevor, and H. I. Christensen. 2010. Semantic Map Partitioning in Indoor Environments using Regional Analysis. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1451–1456.

[16] Andreas Nüchter and Joachim Hertzberg. 2008. Towards Semantic Maps for Mobile Robots. *Robotics and Autonomous Systems* 56, 11 (2008), 915–926.

[17] Dejan Pangercic, Moritz Tenorth, Benjamin Pitzer, and Michael Beetz. 2012. Semantic Object Maps for Robotic Housework - Representation, Acquisition and Use. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vilamoura, Portugal.

[18] Jose L. Part and Oliver Lemon. 2017. Incremental Online Learning of Objects for Robots Operating in Real Environments. In *Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EPIROB)*. Lisbon, Portugal.

[19] Danijel Skočaj, Matej Kristan, Alen Vrečko, Marko Mahnič, Miroslav Janíček, Geert-Jan M Kruijff, Marc Hanheide, Nick Hawes, Thomas Keller, Michael Zillich, et al. 2011. A system for interactive learning in dialogue with a tutor. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 3387–3394.

[20] Danijel Skočaj, Matej Kristan, and Aleš Leonardis. 2009. Formalization of Different Learning Strategies in a Continuous Learning Framework. In *Proceedings of the Ninth International Conference on Epigenetic Robotics; Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, 153–160.

[21] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca A. Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *CoRR* cs.CL/0006023 (2000).

[22] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: an Introduction*. MIT Press.

[23] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J. Mooney. 2017. Opportunistic Active Learning for Grounding Natural Language Descriptions. In *Proceedings of the 1st Annual Conference on Robot Learning (Proceedings of Machine Learning Research)*, Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (Eds.), Vol. 78. PMLR, 67–76.

[24] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL '98)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 271–280.

[25] J. Wu, H. I. Christensen, and J. M. Rehg. 2009. Visual Place Categorization: Problem, dataset, and algorithm. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 4763–4770.

[26] Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2016. Training an Adaptive Dialogue Policy for Interactive Learning of Visually Grounded Word Meanings. In *Proceedings of the SIGDIAL 2016 Conference*. Association for Computational Linguistics, 339–349.

[27] H. Zender, O. Martínez Mozos, P. Jensfelt, G.-J.M. Kruijff, and W. Burgard. 2008. Conceptual Spatial Representations for Indoor Mobile Robots. *Robotics and Autonomous Systems* 56, 6 (2008), 493–502.

[28] Hongwei Zhang, Xiong Xiao, and Osamu Hasegawa. 2014. A Load-Balancing Self-Organizing Incremental Neural Network. *IEEE Transactions on Neural Networks and Learning Systems* 25, 6 (2014), 1096–1105.