# How Much Should a Robot Trust the User Feedback? Analyzing the Impact of Verbal Answers in Active Learning

Victor Gonzalez-Pacheco<sup>1</sup>, Maria Malfaz<sup>1</sup>, Jose Carlos Castillo<sup>1</sup>, Alvaro Castro-Gonzalez<sup>1</sup>, Fernando Alonso-Martín<sup>1</sup>, and Miguel A. Salichs<sup>1</sup>

Universidad Carlos III de Madrid, Leganés, 28914 Madrid, Spain, {vgonzale, mmalfaz, jocastil, acgonzal, famartin, salichs}@ing.uc3m.es, http://roboticslab.uc3m.es

Abstract. This paper assesses how the accuracy in user's answers influence the learning of a social robot when it is trained to recognize poses using Active Learning. We study the performance of a robot trained to recognize the same poses actively and passively and we show that, sometimes, the user might give simplistic answers producing a negative impact on the robot's learning. To reduce this effect, we provide a method based on lowering the trust in the user's responses. We conduct experiments with 24 users, indicating that our method maintains the benefits of AL even when the user answers are not accurate. With this method the robot incorporates domain knowledge from the users, mitigating the impact of low quality answers.

## 1 Introduction

Recent studies in robotics have started to include ideas from Active Learning (AL). Using this kind of learning, robots are able to mimic how humans learn: first, by observing their teacher, and then by asking questions<sup>1</sup> when they have any doubts about the concept to be learnt or the examples they have seen. AL comes from the Machine Learning field and it was introduced by Angluin [3]. Whilst in Passive Learning (PL), the teacher provides the examples to the learner and labels them, in Active Learning it is the learner who takes the initiative by asking queries to the teacher or oracle. The use of AL in robotics has three main motivations when compared with PL. Firstly, active learners can potentially obtain better accuracy of the learned concepts. Secondly, AL may reduce the number of training examples needed to acquire a concept [15]. This is specially relevant in robotics since in interactive learning the cost of acquiring a training example might be time consuming. Finally, people seem to prefer to train robots that learn actively over passive ones [5].

Two research trends can be distinguished in the field. In the first one, robots learn by self-exploring the environment while in the second one, robots leverage

<sup>&</sup>lt;sup>1</sup> Questions and queries will be used indistinctly in this paper.

*HRI* to learn from humans [10]. This paper focuses on the second approach and it is inspired, mainly, by the works of Rosenthal [13] and Cakmak [5,6]. Nevertheless, we attempt to go further by understanding how answers to different types of questions could affect the robot's learning. Rosenthal [13] explores how different questions affect the accuracy and correctness of the user's responses. Cakmak et al. [5] studied how the robot was perceived when it showed three different degrees of interactivity when asking questions. In a related work, Cakmak also [6] studied how humans ask questions when learning.

There is literature that assessed different types of queries for AL [6] and that evaluated how to ask these questions to maximise the accuracy of the user answers [13]. However, we did not found evidence on how different types of queries affect the robot's learning performance. This paper explores the learning impact of different types of queries that seek information regarding the learning parameters in an interactive pose learning task. To do so, we propose a method which consists in reducing the robot's confidence on the user's answers by including more parameters than the user answered.

This paper is divided as follows. First, Section 2 describes our learning approach were we apply AL for pose learning. After that, we present our experiment in Section 3, and the obtained results are shown in Section 4. The results are discussed along with the conclusions in Section 5.

# 2 Learning scheme

When the user starts training the robot, she has to carry out two tasks. The first one is to put herself in the pose she wants to teach the robot. The second task is to tell the robot the pose she is standing at. Gathering the data from vision and from verbal interaction, the robot builds a dataset which feeds a learning algorithm. In this section we describe the components that participate in such learning process.

As a visual input we use the depth data supplied by a Kinect *RGB-D* camera. In order to do this, we employ the OpenNI<sup>2</sup> *API* to build a skeleton model of the user, composed of the positions and orientations of 15 joints of the user (head, neck, torso, shoulders, elbows, hands, hips, knees and feet). This skeleton model is the input for our learning algorithm  $(x_i)$ . The labels  $(y_i)$  of the dataset are gathered interactively during the training process using an Automatic Speech Recognition (ASR) system described in [2].

Once the training ends, the robot applies a learning algorithm, Random Forests [4], to the set of training instances. The algorithm is freely available through the Weka Framework [9]. We use the default parameters as provided by the framework as these provide good performance on a wide range of datasets. Further details about the training process can be found in [8].

<sup>&</sup>lt;sup>2</sup> http://structure.io/openni

#### 2.1 Proposed approach for active learning

We use AL to ask questions related to the poses to be learnt. Active robot learners can ask different types of questions. This paper focuses on *Feature Queries*, which try to find the features of the learning space that are more relevant for learning. *Feature Queries* are perceived by users as the smartest questions [6]. These questions consist in asking the user if a certain feature of the learning space is relevant or not (inspired by [7,12]). In our approach, we ask the questions once the training session is over. At that moment, the robot asks the user which parts of her body have been the most important ones for each pose. For instance, if the user has taught the robot a pointing pose, it is expected that when the robot asks for the features that are more relevant for this pose, the user's answer should indicate some part of her arm.

With the user's answers, the robot filters all the features which has been told to be less relevant. Notice that in this paper we focus on the learning effects of filtering parameters due AL instead of how users perceive the questions or which kind of queries are most helpful. Regarding the questions themselves, we have taken into account the effect in which the user's responses can be affected by the way questions are asked [13]. For that reason, we considered three types of questions to the user: (i) Free Speech Queries (FSQ) consist in open questions which allow the user to answer freely (e.g. "Which is the most important limb in this pose?"); (ii) Yes/No Queries (YNQ), force the user to answer with a yes or no statement (e.g. "Is the hand important?"); and (iii) Rank Queries (RQ), the user must answer quantifying the importance of a limb from *not important* to very important (e.g. "How important is your hand?"). Typically, answers to FSQ provide a single limb or a short list of limbs which the user considered important (e.g. "I suppose my arm"). Therefore, their use might be interesting when the robot does not know which limbs might be important. Conversely, YNQ and RQ force the user to answer only about the limb the robot asked for. Hence, these questions are better to retrieve information from a specific limb, that is, from a specific parameter or set of parameters in the learning space. The major drawback of FSQs is that those answers need to be parsed in order to map what limbs the user has talked about.

Once the robot has all the answers from the user, it processes them to decide what limbs are the most relevant to learn a certain pose. Our system makes this decision through a threshold in which, if a limb has a value below it, it will be filtered out and, therefore, not used for learning. This threshold is calculated differently depending on the type of the questions. In FSQ, each user that has trained the robot provides a list of limbs which she has considered important, so we can calculate the number of times a limb has been mentioned by the users to get a score of its relevance  $R_l^3$ . This score can oscillate between 0, if no one considered that limb important, and  $N_u$  (number of users that have trained the robot) if every user mentioned it when they were asked. Therefore, with the user answers we can build a list of relevances:

<sup>&</sup>lt;sup>3</sup> Note that we use the terms *relevance* and *importance* indistinctly

$$R_{FSQ} = \{R_{head}, R_{neck}, \dots\}$$
(1)

in which are stored the relevances for all the 15 limbs. We then calculate the mean relevance  $\overline{R_{FSQ}}$  of this vector. Our threshold  $Th_{FSQ}$  is established as this mean plus one standard deviation:

$$Th_{FSQ} = \overline{R_{FSQ}} + \sigma_{R_{FSQ}} \tag{2}$$

We decided to add a standard deviation  $\sigma_{R_{FSQ}}$  to the threshold in order to ensure that the limbs which are chosen stand out from the rest. The threshold in YNQ is calculated similarly, but instead of summing the number of times the user mentioned a limb, we sum all the positive answers (the number of times "Yes" was answered). In RQ, the process is slightly different since the answers are not binary (yes/no) but a direct measure of what the perceived relevance of each limb is. Thus, with the answers from several users, we calculated the average relevance for each limb. So  $R_l$  in RQ is actually  $\overline{R_l}$ . The rest of the process is exactly the same as in FSQ and YQ except the criteria in which a limb passes the threshold. In this case, since the relevances are random variables, we decided that a limb passes the threshold if its 95% Confidence Interval (CI) is above it.

With this thresholding mechanism, we can build parameter filters that enable the robot to filter the data which is not relevant for learning. We have built 3 different filters that can be used to pre-process the data before feeding them to the classifier, namely FSQF, YNQF, and RQF (the final F stands for Filter).

Additionally, we created a fourth filter, the *Extended Filter* (EF), which is as an extension of RQF. The EF is an RQF that includes all the adjacent limbs of a normal RQ Filter. For instance, if we have an RQF formed with {head, neck}, its associated EF includes: {head, neck, left shoulder, right shoulder, torso}. As shown in Section 4.2, the EF improves RQF in the situations where, due a low quality answer from the user, the RQF (and other AL filters) are outperformed by Passive Learning. In such cases, the EF behaves as a Passive Learner, while, when the users provide good answers, the EF offers a learning performance comparable to other AL filters.

# 3 Evaluation

The aim of our experiment is to understand how the answers to FSQ, YNQ, and RQ affect robot learning. Accordingly, we prepared an experiment in which several users taught the robot different poses. After the pose acquisition phase, users were asked several questions aimed to improve the robot's learning performance.

Despite the learning session involved natural interaction between the user and the robot, we evaluated the effectiveness of the user's answers in offline questionnaires that were filled to the users just after the training session. This was motivated because the aim of our experiment was to explore the uses of different queries and which ones are better to ask, requiring us to ask many



Fig. 1: Examples of poses that the users taught to the robot. First row: Looking left, forward and right. Second row: Pointing left, forward and right

questions to the users. In that situation, we observed fatigue in the users when the robot asked so many questions to a user in the same experiment.

The user trained the robot Maggie [14], which is equipped with Kinect, ASR [2] and Text to Speech (TTS) systems, coupled in a Natural Dialogue Management System [1], which enable the robot to carry out natural interactions. These components are tightly coupled by using the ROS [11] framework. Although most of these components are self-built, any robot equipped with a Kinect, a TTS and an ASR, can replicate our experiments easily.

#### 3.1 Experimental setup and method

We tested our active learning approach in two experiments where 24 users trained the robot while interacting with it. The 24 users participated in both experiments. Each experiment consisted in the user teaching three poses to the robot: (i) Experiment 01: The users showed the poses *looking left, looking forward, and looking right* (Fig. 1, first row); and (ii) Experiment 02: The users showed the poses *Pointing left, pointing forward, and pointing right* (Fig. 1 second row).

The users were told to stand in front of Maggie. Then, the experimenter explained the experimental procedure and answered any doubts that might have appeared. During training, users had the freedom to start, pause, and finish the training session whenever they wanted. The procedure was: first, the user waited in front of the robot until it tells that it is ready to start. Then, she was able to start the training whenever she wanted. To record each pose, the user was told to: first stand still at this particular pose and then tell the robot the label for that pose. Before changing to other pose, the user had to tell the robot to stop recording this pose. The user interacted with the robot verbally. Once the robot tells the user that it has finished recording the pose, she is free to move to the next pose and start the process again. The user finishes the teaching session telling it to the robot. The session dynamics are similar to a previous work [8].

One dataset per experiment was recorded to feed the learning process. After having trained the robot, they filled a questionnaire where they were asked questions relative to the poses they taught to the robot. These questions were, first FSQ, then YNQ, and finally RQ. All users had to respond to the three types of questions, although their answers were treated separatedly. The following examples of the asked questions are provided<sup>4</sup>: **FSQ**: What parts of your body do you think the robot has used to learn whether you were looking/pointing left, forward or to the right?. Here, users were asked to write in an open-text box what limbs were considered most important in each experiment. YNQ: What parts of your body do you believe the robot has used to learn in the first experiment?. For this set of questions, the user had to fill a multiple-choice list of the limbs depicted in Section 2. **RQ**: Mark the importance of each of the parts of your body so that the robot can learn. RQs consisted in fifteen questions per experiment, each one asking for the importance of a single limb. The answers consisted in a 4-point scale rating the importance of a limb ranging from Not important at all to Very important.

## 4 Results

#### 4.1 Filters for active learning

First, we evaluated the responses of the users to the questionnaires, from which we build the AL filters that will be used for learning. Figure 2 shows the results from the user's answers in each experiment. The horizontal dashed line indicates the filter thresholds. The limbs that passed the threshold are painted in orange, indicating that they will be used in the learning phase.

When analyzing the user's answers to FSQ, we realized that sometimes their answers were too broad. This was the case of the answers to experiment 2, in which most users answered with  $my \ arm$ . There is not a direct mapping between arm and any of the joints. Hence, we decided to include all joints related to the limb, in this case hand, elbow, and shoulder. Moreover, no user indicated which arm was referring when they answered  $my \ arm$ . Thus, we decided that all the uncertain answers that covered different possible limbs would be mapped including all these limbs. For instance, in the case of the  $my \ arm$  answer, we included the user's both arms.

YNQ and RQ produced nearly the same filters, which might occurred because the users had a list of limbs when choosing their answers. Nevertheless, although both graphs have similar shapes, there is a slight difference in the limbs that were not selected. In RQ, these limbs tended to be closer to the threshold than YNQ. For instance, this effect is clearly shown in Fig. 2, first row. These differences

<sup>&</sup>lt;sup>4</sup> The original questions were asked in Spanish. Here we provide the most accurate translations we have found.



Fig. 2: User answers for experiments 01, looking, (first row) and 02, pointing, (second row). The limbs that are selected are painted in orange. Note that in RQ, a limb is selected if its 95% CI is above the threshold (see Section 2).

might have been produced because, although the user had the list of limbs in both cases, in RQ they were forced to give an answer to each limb. Two conclusions might be drawn from these effects: (i) if you let users decide which limbs are important, they provide stricter filters than when you ask them for particular limbs; and (ii) RQ and YNQ obtained the same filters because we were too strict selecting our threshold. Had the threshold been lower, RQ would have had more limbs included than YNQ. In that case RQ could have been considered as an extended version of YNQ.

This idea is what led us to create the Extended Filter presented in Section 2.1. Since the user answers were too simplistic, we decided that the robot should not blindly trust in their answers. However, since they were neither completely wrong, we opted to extend their answers including more information than they actually gave.

#### 4.2 Learning Results

We compared the AL performance when using the filters built from the FSQs, YNQs and RQs against a Passive Learner, presented in [8], including also the EF to the comparison. These filters pre-process the data, removing all the pa-



Fig. 3: Left: Learning Results of Experiment 01: Looking Left, Front, Right. Right: Learning Results of Experiment 02: Pointing Left, Front, Right

rameters associated to the limbs that were filtered out. The remaining data are fed to a random forest classifier [4]. The metric we used for our evaluation is the F1-score.

As already described in Section 4.1, YNQ and RQ produced the same filters, therefore, from now on, we will only describe RQ results. Fig. 3 shows the results for the *looking* (left) and *pointing* (right) experiments. The figure shows how the filters behaved differently in both experiments. In the *Looking* experiment (Fig. 3, left) RQ and FSQ performed worse than PL with the exception of the cases in which only one user trained the robot. However, the EF achieved an F-Score comparable to PL. A different situation occurred in the *Pointing* experiment (Fig. 3, right), where FSQ and RQ scored better results than PL. Here the EFbehaved as well as the other AL queries.

The lower performance of AL in the *looking experiment* might be caused because, in this experiment, the users were providing simplistic answers leading the robot to build filters that omitted relevant data. When the robot had few training examples, the users compensated the lack of data by introducing relevant domain information. However, when the robot gathered enough training examples, their answers prevented the robot to use information it could have helpful for learning. We observed this fact when we checked the learning dataset and found that the data of the user's shoulders might have been relevant. Yet, the users omitted the shoulders when answering the queries, perhaps, because the variations in their shoulders were unperceived.

Because of that, we come up with the idea of lowering the trust in the user's response and including more information than the user is giving. This is exemplified by the EF, which included adjacent limbs to the user's answers. Note that despite the EF seems tailored for our scenario, we believe that a similar approach can be followed in other scenarios where a strong correlation between learning parameters can be found. In that sense, if a user produces a filter including some learning parameters, it might be interesting to include other correlated parameters.

eters as well, even if the user did not include them. Some potential fields that can benefit from this approach are gesture recognition and object recognition among others.

## 5 Conclusions

The main contribution of this paper is twofold. First, we evaluated how different types of Feature Queries affect the learning performance of a robot that learns actively. We found that, in some cases, user's answers might be too simplistic, potentially leading to a reduction in the robot's learning accuracy. In such cases, if the robot trusts too much in the user's responses, Active Learning approaches might be outperformed by Passive Learning. The second contribution of this paper is a method in which the robot reduces its confidence on the user's answers by extending them to other related parameters of the learning space. This method has proven to keep the learning performance high even in the cases where users did not provide accurate answers.

We tested our approach in an experiment where 24 users trained a social robot to recognise poses. Users were asked for three types of queries *Free Speech Queries, Yes/No Queries* and *Rank Queries*. The answers to these queries were used to build feature filters that pre-processed the training data before it was fed to the learning algorithm. We found that, since RQs are more verbose, their filters tend to be more inclusive. However, it was demonstrated that users prefer not to be asked many questions [5], therefore it remains as a future work to explore the optimal balance between a verbose robot and the quality of filters. In this line, we have found that FSQs have the advantage of being more natural and they can be much more efficient than the other types of queries.

From our experiments we concluded that inaccurate user verbal responses in AL may lead to loss of relevant data in the parameter space. When this happens, AL could produce worse results than PL. To solve this problem we developed the notion of the *Extended Filter*. As this filter includes limbs omitted by users, it achieves a performance comparable to PL in the situations where ALdo not worked as well as expected. What is more interesting, when AL performes better than PL, the EF behaves as a regular AL approach. Therefore, our EFgets the best of both worlds, being a good choice when AL is beneficial but it is not possible to control the accuracy in user's answers.

Even though our method is applied for pose learning, we believe that it could be applied to other AL-based learning approaches. This is because our approach is based on feature selection, Hence, other learning approaches might apply it as long as the robot knows the relationship between the features it is asking for.

## Acknowledgment

This research has received funding from the projects *Development of social robots* to help seniors with cognitive impairment - ROBSEN funded by the Ministerio de Economía y Competitividad (DPI2014-57684-R) from the Spanish Government; and *RoboCity2030-III-CM* (S2013/ MIT-2748), funded by Programas de Actividades I+D of the Madrid Regional Authority and cofunded by Structural Funds of the EU.

## References

- Alonso, F., Gorostiza, J., Salichs, M.: Preliminary Experiments on HRI for improvement the Robotic Dialog System (RDS). In: Robocity2030 11th Workshop on Social Robots (2013)
- Alonso-Martín, F., Salichs, M.A.: Integration of a voice recognition system in a socia robot. Cybernetics and Systems 42(4), 215–245 (May 2011)
- 3. Angluin, D.: Queries and concept learning. Machine Learning 2(4), 319-342 (1988)
- 4. Breiman, L.: Random Forests. Mach. Learn. 45(1), 5-32 (2001)
- Cakmak, M., Chao, C., Thomaz, A.L.: Designing Interactions for Robot Active Learners. IEEE Trans. on Autonomous Mental Development 2(2), 108–118 (2010)
- Cakmak, M., Thomaz, A.L.: Designing robot learners that ask good questions. In: Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12. p. 17. ACM Press, New York, New York, USA (2012)
- Druck, G., Settles, B., McCallum, A.: Active learning by labeling features. In: Proc. of the 2009 Conf. on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. pp. 81–90. Association for Computational Linguistics (2009)
- Gonzalez-Pacheco, V., Malfaz, M., Fernandez, F., Salichs, M.A.: Teaching human poses interactively to a social robot. Sensors 13(9), 12406–12430 (2013)
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
- Lopes, M., Oudeyer, P.Y.: Guest Editorial Active Learning and Intrinsically Motivated Exploration in Robots: Advances and Challenges. IEEE Transactions on Autonomous Mental Development 2(2), 65–69 (Jun 2010)
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source Robot Operating System. In: Open-Source SW Workshop of the Int. Conf. on Robotics and Automation (ICRA) (2009)
- Raghavan, H., Madani, O., Jones, R.: Active Learning with Feedback on Features and Instances. J. Mach. Learn. Res. 7, 1655–1686 (2006)
- Rosenthal, S., Dey, A.K., Veloso, M.: How robots' questions affect the accuracy of the human responses. In: RO-MAN 2009 - The 18th IEEE Int. Symposium on Robot and Human Interactive Communication. pp. 1137–1142. IEEE (2009)
- Salichs, M., Barber, R., Khamis, A., Malfaz, M., Gorostiza, J., Pacheco, R., Rivas, R., Corrales, A., Delgado, E., Garcia, D.: Maggie: A Robotic Platform for Human-Robot Social Interaction. In: 2006 IEEE Conference on Robotics, Automation and Mechatronics. pp. 1–7. IEEE, Bangkok (Dec 2006)
- Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2010)