In the *Autonomous Robots and Multirobot Systems* (ARMS) Workshop at AAMAS, Istanbul, Turkey, May 4, 2015.

Incremental Knowledge Acquisition for Human-Robot Collaboration

Batbold Myagmarjav[1] and Mohan Sridharan[2]

[1] Department of Computer Science,
Texas Tech University, Lubbock TX 79409, USA
bat.myagmarjav@ttu.edu,
[2] Department of Electrical and Computer Engineering
The University of Auckland, Auckland 1142, New Zealand
m.sridharan@auckland.ac.nz

Abstract. Robots require a significant amount of domain knowledge to collaborate with humans in complex domains. Since it is difficult to provide accurate and complete domain knowledge, active learning algorithms have been developed to enable robots to acquire relevant information by posing questions when necessary. Human participants may, however, lack the time and expertise to provide elaborate and accurate responses. Success of active learning in human-robot interaction thus depends on robots posing questions that enable faster learning using limited interaction with non-expert humans. Towards this objective, this paper presents an architecture for incremental active learning. Robots equipped with this architecture construct candidate questions using local and global contextual cues. These queries are ranked based on utility, which is is measured as a combination of measures for ambiguity and information gain. Human responses to the top-ranked questions are used to update the robot's knowledge. This paper illustrates and evaluates the architecture's capabilities in a simulated domain, significantly reducing the number of questions posed in comparison with algorithms that use the individual measures or a strategy for randomly selecting questions.

Keywords: Human-robot interaction, incremental knowledge acquisition, contextual query generation.

1 Introduction

Robots[3] need a significant amount of domain knowledge to collaborate with humans in complex application domains. Since it is difficult to equip robots with accurate and complete domain knowledge, robots frequently have to solicit help from humans to perform the desired tasks. However, it is often the case that human availability is scarce, and the human participants lack the expertise to provide elaborate instructions to robots. The ability to pose relevant questions

[3] Terms "agent", "robot" and "learner" are used interchangeably in this paper.

that quickly draw a human's attention to the object(s) of interest can thus significantly influence the quality of a robot's interaction with humans.

Humans frequently use contextual cues to draw attention to an object of interest. Such contextual information is all the more useful when the word(s) we use to describe an object are different from those used by our collaborator, or if our collaborator does not have the knowledge necessary to understand our description. Contextual cues can take different forms, and positional context with reference to a known object can be very useful in disambiguating the object of interest. For instance, instead of referring to a "1965 Ford Mustang" in a busy street intersection, we may refer to the "red car behind the bus", using feature labels (e.g., color and object labels) and positional reference to a known object. Humans also incrementally learn from, and build upon, existing knowledge, by posing questions to acquire information from parents, teachers and friends. Furthermore, we attempt to formulate interesting questions that help us quickly acquire the desired information. Consider, for instance, the common question: "what is that?", which even in the presence of other cues (e.g., gestures) is likely to provide an ambiguous reference to the person we are interacting with, resulting in a possibly inaccurate response. In contrast, the question: "what is in your right hand?" is more likely to obtain an accurate (and useful) answer by unambiguously drawing attention to the object of interest. Motivated by these instinctual choices made by humans, this paper describes an architecture for incremental knowledge acquisition in human-robot interaction using visual and verbal cues. The architecture enables robots to:

- Form candidate questions about a scene under consideration based on an analysis of the domain knowledge, and the local and global contextual information currently available for use.
- Rank candidate questions in decreasing order of relative utility, with utility being computed using heuristic measures of information gain, ambiguity and human confusion.
- Solicit human feedback by posing top-ranked questions, and use human responses to incrementally update knowledge about properties of objects in the scene under consideration.

In this paper, we illustrate and evaluate these capabilities of the proposed architecture using simulated images of scenes with objects of different colors, shapes, and sizes. The robot's objective is to start with incomplete knowledge and learn the labels of all the objects and features in the scene by asking as few questions as possible. This choice of objective and domain corresponds to a "thought experiment" that allows us to control the related factors, and analyze the contributions of the proposed algorithms and measures.

The remainder of the paper is organized as follows. Section 2 motivates the proposed architecture by briefly reviewing a representative set of related work. Section 3 describes the proposed architecture and its components. Section 4 describes the experimental setup and discusses the results of experimental evaluation. Finally, Section 5 presents the conclusions along with future plans.

2 Related Work

This section motivates the proposed architecture by reviewing a representative set of related work in active learning and human-robot interaction (HRI).

Researchers have designed many active learning algorithms to minimize the training data required in comparison with classical supervised learning algorithms. These algorithms allow incremental labeling or acquisition of data, e.g., by allowing a human *annotator* to label instances in the data set that have been misclassified using existing models. A recent survey categorized active learning algorithms into *pool-based*, *stream-based* and *membership query* algorithms [9]. Existing algorithms predominantly focus on choosing unlabeled instances that are to be presented to the annotator, rather than evaluating the types of queries to ask [9], [12], [13]. However, research indicates that query type influences the information obtained, e.g., the use of queries about labels of feature and object instances significantly improves object recognition based on the models learned [6].

Active learning has been combined with multiple instance learning (MIL) to enable more effective HRI, minimizing human supervision by supporting the labeling of bags (e.g., images) instead of individual instances (e.g., objects and features in the images) [10]. Research shows that even when active learning is combined with MIL, an incremental learning architecture that adds the ability to solicit labels of previously unseen bags results in much faster learning of object models and more accurate object recognition based on the learned models [8]. Research also shows that a multimodal learning algorithm that associates visual features with verbal descriptions (provided by humans) leads to object models that result in more accurate object recognition than models based on just visual features [11]. Although these algorithms reduce human involvement in the learning process, the queries being posed focus on labeling bags and do not pay attention to the type of queries being posed.

Artificial intelligence and robotics researchers have developed algorithms that allow learning agents to ask different types of questions. For instance, context has been embedded in questions to improve the overall quality of the questions being posed in an HRI setting [7]. However, this approach focused on the reaction of humans to these questions, and the ability of humans to answer these questions correctly, but not on the agent's ability to learn from these questions. Another approach for asking the right questions developed a decision tree with the objective of identifying a series of questions that would extract the desired information [4]. Posing query generation as a planning task requires prior knowledge of possible answers, which will be different for different scenes; the planning will also be computationally inefficient.

Learning from demonstration (LfD) algorithms allow agents to observe a human teacher demonstrate a specific task, and either mimic the observed actions or map the actions to the available capabilities. Common algorithms that use *teleoperation*, *planning* and *demonstration* learning techniques have been surveyed and discussed in [1], [2]. More recent research has combined active learning with LfD to explore the use of four types of questions: *object label*, *feature label*, *demonstration*, and *affirmation* queries [3]. However, the objective was to

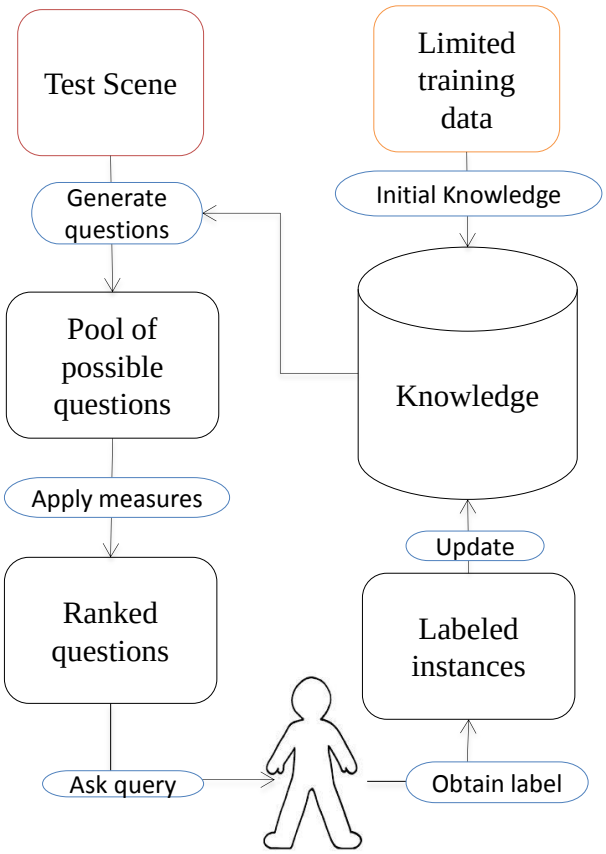


Fig. 1. Proposed architecture for incremental knowledge acquisition.

explore how each query category is perceived by the human, i.e., if the human thinks that the robot asked a "smart question" in specific situations.

This paper seeks to build on and address the limitations of existing work. Our architecture allows the agent (i.e., the learner) to use contextual cues and incrementally pose questions with high relative utility, i.e., questions that help disambiguate between, and quickly acquire information about, domain objects.

3 Problem Formulation

Figure 1 is an overview of the proposed architecture in the context of images of scenes with objects with different properties (e.g., color, shape and size). The architecture starts with limited knowledge of the *scene*. The set of possible queries is generated as described in Section 3.1. Based on measures of information gain, ambiguity, and human confusion, (Sections 3.2.1-3.2.3), the most useful queries are selected to solicit input from a human participant, as described in Section 3.2.

Once human input (i.e., annotation) is obtained, the domain knowledge is updated and used to generate subsequent queries until all objects and features are labeled. We will use the following notation throughout this paper:

1. An object can be characterized by n different properties or *features*.
2. A superset of features is denoted by $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$. Each $\mathcal{F}_i$ is a set of instances of one type of feature (e.g., color) where $1 \leq i \leq n$.
3. A feature instance $f \in \mathcal{F}_i$ consists of tuple $\langle label, values \rangle$, where *label* is a human understandable word, e.g., *red*, and *values* refer to the quantitative representation of that label, e.g., RGB value $(255, 0, 0)$.
4. A *Scene* consists of a set $\mathcal{S}$ of objects. Each object $s \in \mathcal{S}$ is denoted by $\langle label, \mathcal{OF} \rangle$, where $\mathcal{OF} = \{f_1, f_2, ..., f_n\}$ with $f_i \in \mathcal{F}_i$ and $1 \leq i \leq n$, i.e., each object has a label and one instance of each possible feature. An object s with feature f_i is denoted by $f_i(s)$. Figure 3 shows some example *scenes*.
5. A set of *Relations* that can exist between two objects in a scene is denoted by $\mathcal{R}$. Each relation $r \in \mathcal{R}$ is assumed to be determinable, and each object is only assigned labels that are known to the architecture. Examples include the relative positions of two objects, and the temporal relation between two events. Such relations are denoted by $r(s_i, s_j)$, where $s_i, s_j \in \mathcal{S}$ and $s_i \neq s_j$.
6. A *Knowledge Base* $\mathcal{K}$ is the tuple $\langle \mathcal{S}, \mathcal{LS}, \mathcal{US}, \mathcal{LF}, \mathcal{UF} \rangle$, where:
   - $\mathcal{LS}$ denotes the set of labeled scene objects such that $\mathcal{LS} \subseteq \mathcal{S}$.
   - $\mathcal{US}$ denotes the set of unlabeled scene objects such that $\mathcal{US} \subseteq \mathcal{S}$. Note that $\mathcal{LS} \cap \mathcal{US} \equiv \varnothing$, i.e., no common members exists in these sets.
   - $\mathcal{LF}$ denotes the superset of labeled features: $\{\mathcal{LF}_1, \mathcal{LF}_2, ..., \mathcal{LF}_n\}$. Each set $\mathcal{LF}_i \subseteq \mathcal{F}_i$ contains the instances of features with labels with $1 \leq i \leq n$.
   - $\mathcal{UF}$ denotes the superset of unlabeled features: $\{\mathcal{UF}_1, \mathcal{UF}_2, ..., \mathcal{UF}_n\}$. Each set $\mathcal{UF}_i \subseteq \mathcal{F}_i$ contains the instances of features with labels, where $1 \leq i \leq n$. Note that $\mathcal{LF}_i \cap \mathcal{UF}_i \equiv \varnothing$.

This notation is used below to describe the generation of candidate queries (Section 3.1), and the use of heuristic measures to rank queries to be posed to human participants (Section 3.2).

3.1 Query Generation

This section describes the generation of a set of candidate queries $\mathcal{Q}$ for a scene, where each query $q \in \mathcal{Q}$ contains embedded contextual information to describe the object of interest. Specifically $q = \langle t, s, \mathcal{C} \rangle$, where:

- t denotes the query type. Specifically, t can indicate that the query under consideration is an object label query or feature label query.
- $s \in \mathcal{S}$ denotes the object of interest in the scene.
- $\mathcal{C}$ denotes the embedded context which describes the object of interest. Specifically $\mathcal{C} = \langle \mathcal{SC}, \mathcal{LC}, gc \rangle$ where:
  - $\mathcal{SC}$ denotes the set of *self contexts* embedded in the query. The labeled feature(s) of s or the label of s can be a self context.

- $\mathcal{LC}$ denotes the set of *local contexts* embedded in the query. Local contexts are defined as labeled objects or features that are related to the object of interest s. In other words, we consider $r(s, s_i)$ such that $s_i \in \mathcal{S}$, $r \in \mathcal{R}$, $s_i \neq s$, with $s_i \in \mathcal{LS}$ or $\exists f \in \mathcal{LF}$ such that $f(s_i)$.
- gc denotes the *global context* embedded in the query. Global context is defined by its relation to the whole scene. Only one global context is assumed to exist for each object, e.g., an object may be in the *top right corner* of the scene. It is also assumed that an object's relation to the scene is computable and the label of each such relation is known.

We consider different levels of contextual information, and Algorithm 1 describes the generation of queries with *level 1* context. For a specific scene object s provided as input, the output is a set of possible questions $\mathcal{Q}$. First, all the context which can describe s is retrieved. Global context gc is assumed to be computed by predefined subroutines and differs depending on the domain. Position relationships are computed in a coordinate system, while event relationships are calculated based on time values. Self contexts $\mathcal{SC}$ of s are the known label or labeled features of s (e.g., *red*). Local contexts are the labeled objects or the objects with labeled features that are related to s through a known relation (e.g., *above* red object). First, if the object label is unknown, i.e., $s \in \mathcal{US}$, an object label query is generated using the global context to s and added to the query set. Object label queries are also generated using each of the self contexts of s, and using each of element of the local context of s, and added to $\mathcal{Q}$. After the object queries are generated, each feature of s is checked for labels. Using the same global context used above, each unlabeled feature generates a feature label query to be added to $\mathcal{Q}$. Feature label queries are also generated using the self contexts of s, and using the local contexts of s, and added to $\mathcal{Q}$, which is returned as output. Note that if no known context exists to describe s, then no candidate queries will be generated about the unlabeled components of s.

A simplistic question template was used for constructing questions:

```
<Question word> <Type> <Context>?
<What is the> <label of the color> <below the cross>?
```

Self context information, e.g., *red object*, is an exception to this template.

3.1.1 Level of Context While contextual cues are useful, humans can be overwhelmed by a large amount of contextual information, especially if they do not have domain expertise. Since this information overload can result in inaccurate responses, we consider different levels of contextual information, and limit ourselves to three levels.

We use α to denote *human confusion*, and introduce a simple measure of this confusion later in this paper. **Level 1** queries are least likely to confuse the human annotator, while **Level 3** queries are the most confusing due to the amount of contextual information considered.

– **Level 1**: One item of contextual information.

Algorithm 1: Level 1 Query Generation

Input: s: a scene object, and knowledge base
Output: $\mathcal{Q}$: set of queries

Procedure *QueryGeneration()*
$\mathcal{C} \leftarrow Context(s)$
Initialize $\mathcal{SC}$ *with* $\mathcal{C}[0]$
Initialize $\mathcal{LC}$ *with* $\mathcal{C}[1]$
Initialize gc *with* $\mathcal{C}[2]$
if $s \in \mathcal{US}$ **then**
  $q \leftarrow \langle object, s, \langle \varnothing, \varnothing, gc \rangle \rangle$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
  for ***each*** $sc \in \mathcal{SC}$ **do**
    $q \leftarrow \langle object, s, \langle \{sc\}, \varnothing, null \rangle \rangle$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
  end
  for ***each*** $lc \in \mathcal{LC}$ **do**
    $q \leftarrow \langle object, s, \langle \varnothing, \{lc\}, null \rangle \rangle$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
  end
end
for ***each*** *feature* f *in* $f(s)$ **do**
  if $f \in \mathcal{UF}$ **then**
    $q \leftarrow \langle f, s, \langle \varnothing, \varnothing, gc \rangle \rangle$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    for ***each*** $sc \in \mathcal{SC}$ **do**
      $q \leftarrow \langle f, s, \langle \{sc\}, \varnothing, null \rangle \rangle$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    end
    for ***each*** $lc \in \mathcal{LC}$ **do**
      $q \leftarrow \langle f, s, \langle \varnothing, \{lc\}, null \rangle \rangle$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    end
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
  end
end
return $\mathcal{Q}$

Procedure *Context(s)*
$\mathcal{SC} = \{f \in \mathcal{LF} \mid f(s)\}$
if $s \in \mathcal{LS}$ **then**
  $\mathcal{SC} = \mathcal{SC} \cup \{label\ of\ s\}$
end
$\mathcal{LC} = \{s_i \in \mathcal{S} \mid s_i \neq s, \exists r(s, s_i), s_i \in \mathcal{LS}\}$
$\mathcal{LC} = \mathcal{LC} \cup \{s_i \in \mathcal{S} \mid s_i \neq s, \exists r(s, s_i), f(s_i) : f \in \mathcal{LF}\}$
Compute $gc(s)$ ▷ predefined subroutine
return $\langle \mathcal{SC}, \mathcal{LC}, gc \rangle$

  - One self context, e.g., *red* object.
  - Two self contexts, e.g., *red rectangular* object.
  - One local context, e.g., object *above the red object*.
  - One global context, e.g., *top right corner* of the scene.
- **Level 2**: Two items of contextual information.
  - One self context and one global context, e.g., *red* object in the *top right corner*.
  - One self context and one local context, e.g., *red* object *above the triangle*.
  - One local context and one global context, e.g., object *above the red object at the top* of the scene.
  - Two local contexts, e.g., object *above the red object* and *right of the circle*.
- **Level 3**: Three items of contextual information.
  - One self context, one global context, and one local context, e.g., *red* object *in top right corner*. *next to the* circle.)
  - Two self contexts and one local context. (e.g., *red circular* object *above the triangle*)
  - Two self contexts and one global context. (e.g., *red circular* object *at the top* of the scene)
  - Two local contexts and one self context. (e.g., *red* object *right of* the circle and *above* the green object.)
  - Two local contexts and one global context. (e.g., object on the *right* of the scene, *above the* circle and *right of* the yellow object)

Queries of a specific level are generated as long as the corresponding contextual cue exists, i.e., there are labeled feature or object instances that can be used to describe the object of interest.

3.2 Query selection

After the set of queries $\mathcal{Q}$ is generated, the most useful queries are selected for annotation. Algorithm 2 describes the steps involved in ranking and selecting the best query based on three heuristic measures. Intuitively, a robot interacting with a human should pose questions that: (1) maximize information gain; (2) minimize ambiguity; and (3) minimize human confusion. We designed heuristic measures based on these intuitive principles. The first measure captures the potential information gain if human annotation is obtained for a query q (Section 3.2.1). The second measure captures how the embedded contextual information in q uniquely describes the object of interest (Section 3.2.2). The information obtained from these two measures is combined, using a measure of human confusion to break ties (Section 3.2.3).

3.2.1 Information Gain: Consider the situation in which the robot has a set of m distinct objects in the scene, and each object has one instance of each feature being considered. For instance, if objects characterized by color and shape features, an object can have *blue* and *rectangle* as the color and shape

Algorithm 2: Query Selection

Input: s: a scene object
Output: q: selected query

```
for each q ∈ Q do
    Compute β                          ▷ Section 3.2.1
    Compute γ                          ▷ Section 3.2.2
    Compute δ                          ▷ Section 3.2.3
    δ(q) ← δ
end
return q_i ∈ Q | argmax_i δ(q_i)
```

feature values, i.e., $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$ such that $\mathcal{F}_1 = \{color\}$, $\mathcal{F}_2 = \{shape\}$, and $\mathcal{F}_1 \ni f = blue$ and $\mathcal{F}_2 \ni f = rectangle$. The sum of the number of labeled and unlabeled feature instances in each feature set equals the number of objects in the scene. The sum of the number of labeled objects and unlabeled objects satisfies the same constraint:

$$|\mathcal{LF}_i| + |\mathcal{UF}_i| = |\mathcal{LS}| + |\mathcal{US}| = |\mathcal{S}| = m$$

for each $1 \leq i \leq n$. The ratio of the number of instances of each feature or object the learner knows against the potential knowledge the learner can acquire is denoted by:

- $P(\mathcal{F}_i) = \frac{|\mathcal{LF}_i|}{m}$
- $P(\mathcal{S}) = \frac{|\mathcal{LS}|}{m}$

Formulated in this fashion, the overall information gain (β) can be measured as the product of quantities computed above:

$$\beta = \prod_{i=1}^{n} P(\mathcal{F}_i) \times P(\mathcal{S})$$

The potential *information gain* from a candidate query q is thus obtained by computing β considering the feature and object instances likely to be labeled upon receiving the answer to q from a human.

3.2.2 Unambiguity: The second measure evaluates a query based on the extent to which the contextual information embedded in the query uniquely identifies an object. As the number of scene objects that satisfy the contextual information embedded in a query increases, the query becomes more ambiguous.

If an interesting object or feature is identified, the learner must also determine how much contextual information it should provide in order to get an accurate response from the human. An accurate response will help minimize the effort wasted in unnecessary interaction. For instance, consider the difference between following two queries:

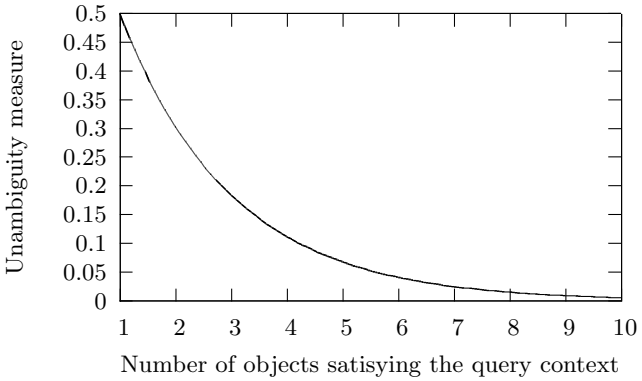


Fig. 2. Unambiguity measure as a function of the number of objects satisfying the context embedded in a query; queries that match multiple objects are more ambiguous.

- "What is the label of the red object?"
- "What is the label of the red object above the bottle?"

The difference in these two queries is the level of contextual information embedded in them; the second query is less ambiguous than the first query.

We use a modified Chi-square probability distribution with degree of freedom $k = 2$ to model the unambiguity measure γ, with $x \in [0, +\infty]$ denoting the number of objects or feature instances in the scene which satisfy the contextual information embedded in a candidate query:

$$\gamma = f(x) = \begin{cases} x = 0, & 0 \\ x \geq 1, & \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})}(x-1)^{\frac{k}{2}-1}e^{-\frac{x-1}{2}} \end{cases}$$

Figure 2 illustrates the distribution that can be simplified for $k = 2$ as:

$$\gamma = \frac{1}{2\Gamma(1)}e^{-\frac{x-1}{2}}$$

where Γ is the Gamma function. Further simplification of the function with $\Gamma(1) = 1$ yields:

$$\gamma = \frac{1}{2}e^{-\frac{x-1}{2}}$$

This mathematical representation captures the desired intuition: as the number of objects satisfying the context embedded in a query increases, the query becomes increasingly less unambiguous.

3.2.3 Combined score: As stated earlier, candidate queries will be ranked in decreasing order of *utility* δ, which is based on measures of potential information gain, unambiguity, and human confusion. The δ of each candidate query is first computed as the product of the two measures described above, i.e., $\delta = \beta \times \gamma$.

This score δ is used to rank the query relative to the other queries; a query with a higher δ is preferred for soliciting information from a human. If there are multiple queries with the same δ, a measure of human confusion (α) is used to break the tie; a query with a lower value of α is preferred. Human confusion is computed based on the level of context embedded in the query, e.g., *Level 1* queries, i.e., queries with the least amount of context embedded in them, will be assigned lower values (of α) than *Level 2* queries, which, in turn, will be assigned lower values than *Level 3* queries. Section 3.1.1 describes these levels of context in detail. The α measure captures the intuition that as the amount of contextual information embedded in a query increases, the query is more likely to confuse the human, and its overall utility decreases. If multiple queries still have the same overall score, one of these queries will be selected randomly.

4 Experimental Results

This section describes the experimental setup (Section 4.1) and summarizes the results of experimentally evaluating the algorithms described above (Section 4.2).

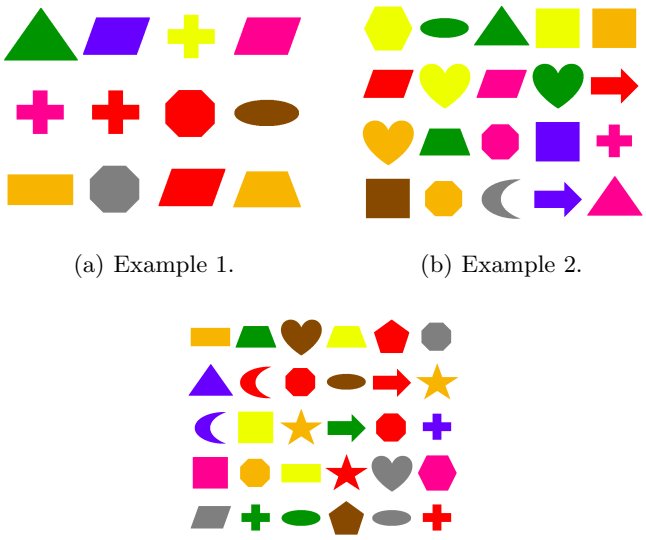
We report results of evaluating our architecture in a simulated domain[4]. The simulated domain allows us to analyze the contributions of the individual measures by controlling the associated factors. The simulated domain abstracts away the uncertainty that exists when object recognition and speech understanding algorithms are applied to visual and verbal cues (respectively). For instance, we assume that an object can be recognized once the models necessary to identify the object's individual features are learned, and that human speech gets translated into text that is parsed to extract the necessary labels. Furthermore, in the experimental results below, we primarily conducted trials with simulated images of scenes, with objects characterized by color and shape features. The labels of interest therefore include the color labels, shape labels, and object labels[5].

4.1 Experimental Setup

For objects characterized by specific colors and shapes, the feature set consists of $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$, where $\mathcal{F}_1 \ni f = \langle label, RGB \rangle$ is a tuple of RGB values and color labels. Ten different basic colors are considered in the experimental trials: *Blue, Brown, Grey, Green, Orange, Pink, Red, Yellow, White* and *Black*. The representation for colors *White* and *Black* is assumed to be always known; they constitute the foreground and background colors. Next, $\mathcal{F}_2 \ni f = \langle label, contour \rangle$ is a tuple of labels and shape contour information; a contour is represented as a set of points on a plane. The 15 shapes in the domain are: *Arrow, Circle, Cross, Heart, Hexagon, Moon, Octagon, Oval, Parallelogram, Pentagon, Rectangle, Square, Star, Trapezoid, Triangle*. A scene in an experimental trial is thus

[4] Preliminary results have been summarized in an extended abstract that will be presented in the main conference [5].

[5] In the examples below, object labels are a combination of the color and shape labels, but this is not a requirement.



(a) Example 1. (b) Example 2.

(c) Example 3.

Fig. 3. Examples of scenes used in the experimental trials.

a set of colored shapes placed without any occlusion; three examples of such scenes are shown in Figure 3.

The set of relations $\mathcal{R}$ considered in this simulated domain denote spatial relationships. It is assumed that the robot is aware of the x and y coordinates of the centroid of an object's shape; the centroid is an important position feature of the object. Relations $r \in \mathcal{R}$ are assumed to be known (i.e., given) to the robot; they can be one of the following:

- *above/up/on top* (centroids' relative locations along the y axis),
- *below/under/beneath* (centroids' relative locations along the y axis),
- *left of/next to* (centroids' relative locations along the x axis),
- *right of/next to* (centroids' relative locations along the x axis),

The experimental setup allows no more than two spatial relationships to exist between any two objects. The objective of the learner is to learn the labels of the objects in the scene as well as labels of the features present in the scene. The proposed architecture, comprising the algorithms and measures described above, is designed to complete this task by posing as few questions as possible. Therefore, the number of questions posed is used as a performance measure.

4.2 Experimental Results

Consider two illustrative examples of query generation. First, consider the scene in Figure 3(a), and assume that the robot's initial knowledge includes the color

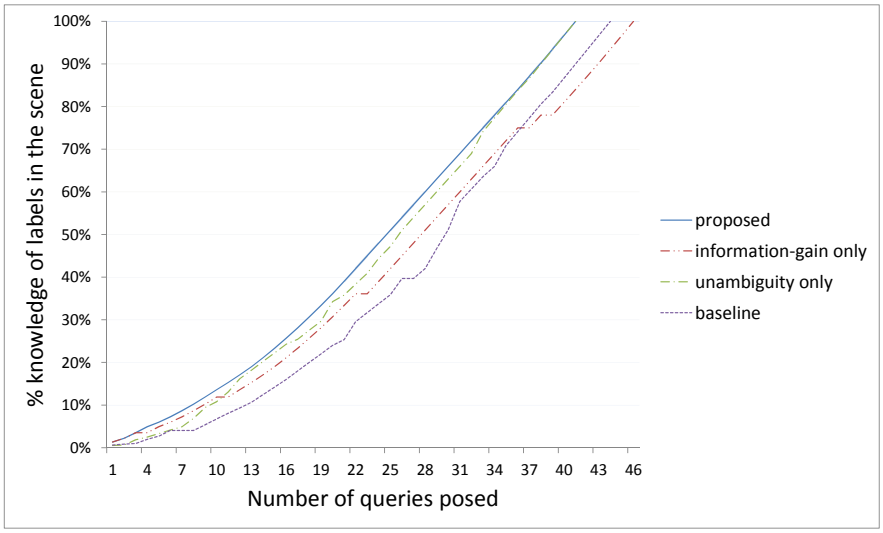


Fig. 4. Knowledge of object and scene labels expressed as a function of the number of queries posed to obtain this knowledge, for the scene in Figure 3(c). The proposed algorithm acquires knowledge faster than algorithms that use just the information gain measure or the unambiguity measure, or select questions randomly.

labels, shape labels, and object labels of the following four objects: *pink star*, *green arrow*, *blue heart*, and *yellow cross*; not all these objects exist in the scene in Figure 3(a). The following are a subset of the questions generated by the system; each line starts with the iteration number and ends with the answer provided to the question:

- *Iteration 4:* "What is the label of the object in the bottom right of the scene?" **Orange Trapezoid**.
- *Iteration 6:* "What is the label of the object that is to the left of the orange trapezoid?" **Red Parallel**.
- *Iteration 13:* "What is the label of the object that is above the red parallel?" **Red Octagon**.

As another example, consider the scene in Figure 3(b), and assume that the robot's initial knowledge includes the color labels, shape labels, and object labels of the same set of four objects as in the previous example. A subset of the questions posed by our system are listed below:

- *Iteration 1:* "What is the label of the object that is to the left of the pink cross?" **Purple Square**.
- *Iteration 15:* "What is the label of the object that is above the purple square?" **Green Heart**.

In both examples, the information obtained by posing questions is used to formulate and pose questions in the subsequent iterations. Note that the questions

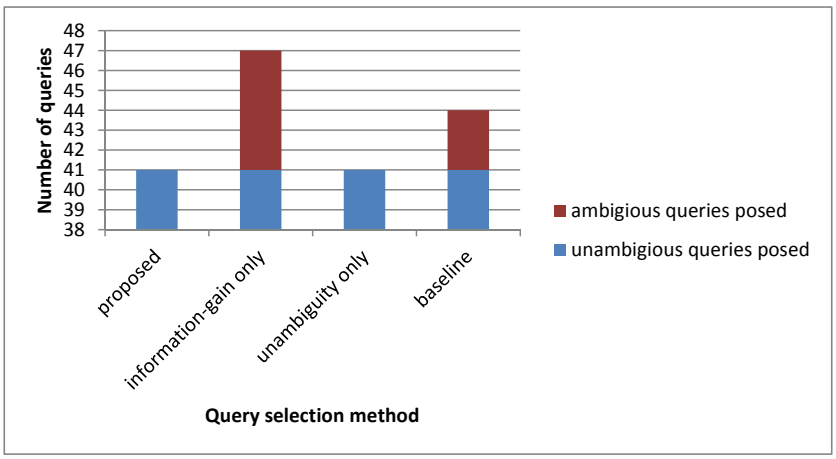


Fig. 5. Number of queries required to acquire the labels of all the objects and features in the scene in Figure 3(c), for the algorithms being compared. Eliminating ambiguous queries significantly decreases the number of queries posed.

may refer to more than one object. Overall, the system incrementally obtains the necessary information by building on the existing knowledge.

Next, Figure 4 compares the proposed algorithm for ranking and selecting queries with three other algorithms for the scene illustrated in Figure 3(c): (1) using only the information gain measure; (2) using only the unambiguity measure; and (3) a baseline approach that randomly selects queries from the candidate set. Figure 4 plots the % knowledge of object and feature labels in the scene as a function of the number of queries posed to acquire this knowledge. Since our proposed algorithm combines *information-gain* and *unambiguity measure* to select high utility queries from $\mathcal{Q}$, it provides the best performance. In contrast, the baseline approach chooses queries randomly from $\mathcal{Q}$, and requires the maximum number of queries to acquire knowledge of object and feature labels in the scene. If an ambiguous query is posed to the annotator, the interaction is considered unsuccessful and leads to no answer. This allows the query selection algorithm that only uses the *unambiguity measure* (see Section 3.2.2) to obtain complete knowledge of the scene by posing the same (total) number of queries as our proposed algorithm. However, the proposed algorithm allows the robot to maximize the amount of knowledge (about the scene) acquired during each interaction with a human in the intermediate stages. Since the algorithm that only uses the information gain measure poses ambiguous queries (similar to the random query selection algorithm), it often results in unsuccessful interactions. The performance improvement provided by the proposed algorithm is likely to be more pronounced in more complex scenes, especially when the uncertainty in sensor input processing is not abstracted away.

Figure 5 summarizes the number of ambiguous and unambiguous queries posed by each of the four query selection algorithms. We observe that eliminating

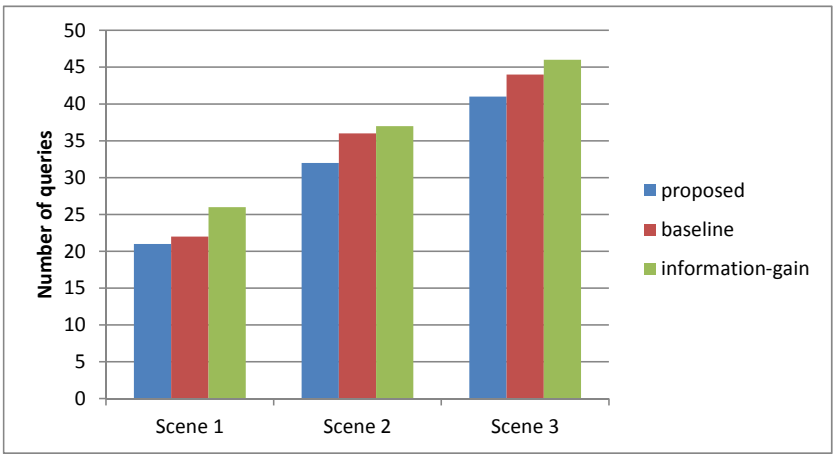


Fig. 6. Number of questions required to learn the color, shape and object labels in three different scenes. Proposed algorithm requires significantly fewer number of queries than an algorithm that only uses the information gain measure, or a baseline algorithm that select queries randomly.

ambiguous queries can significantly decrease the number of queries posed to acquire knowledge of object and feature labels in the scene. Finally, Figure 6 summarizes the number of queries posed for the three scenes in Figure 3, which differ in terms of the number and type of objects; *Scene 1*, *Scene 2* and *Scene 3* have 12, 20 and 30 objects respectively. For each scene, the robot started with the same initial knowledge about a subset of objects in the scene, i.e., labels of these objects and their color and shape features. The selection of questions from the set of candidate questions $\mathcal{Q}$ was based on the proposed algorithm (see Section 2). As the baseline for comparison, we used an algorithm that started with the same initial knowledge but selected queries randomly from $\mathcal{Q}$. For each set of paired experimental trials, our algorithm results in the robot learning the desired labels of objects and features in different scenes by posing a much smaller number of queries. Similar results were obtained over 100 randomly generated scenes with different number and type of objects.

5 Conclusion and Future Work

Robots typically need a significant amount of domain knowledge to collaborate with humans in practical domains. However, it is difficult to equip robots with accurate and complete domain knowledge, and humans may not have the time and expertise to provide elaborate feedback. The architecture described in this paper builds on, and significantly extends, the existing work in active learning. The architecture generates candidate queries based on contextual information, and combines heuristic measures of information gain, ambiguity, and human confusion, to rank queries based on their relative utility. Top-ranked queries are used

to solicit human feedback, and the responses are used to incrementally guide the selection of the subsequent queries. Experimental results in a simulated domain with scenes of objects characterized by colors and shapes show that the proposed approach significantly reduces the number of queries posed in comparison with algorithms that use the individual measures, or an algorithm that selects questions randomly.

Our architecture opens up many directions for further research. First, other types of queries based on contextual information can be explored. We also plan to evaluate our architecture on more complex scenes with other types of objects. Another direction of future research is to implement and evaluate the algorithm on physical robots in the presence of non-deterministic actions and noisy observations. Finally, the current work assumes that human response is accurate, which is not always the case; one direction of future research is to explicitly model and account for the uncertainty in the response provided by the humans. The long-term objective of this research is to enable human-robot collaboration in complex application domains.

References

1. B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009.
2. E. A. Billing and T. Hellstrom. A formalism for learning from demonstration. *Journal of Behavioral Robotics*, 2010.
3. M. Cakmak and A. Thomaz. Designing robot learners that ask good questions. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 17–24, March 5-8, 2012.
4. M. Gervasio, E. Yeh, and K. Myers. Learning to ask the right questions to help a learner learn. In *ACM International Conference on Intelligent User Interfaces*, pages 135–144, Palo Alto, USA, February 13-16, 2011.
5. B. Myagmarjav and M. Sridharan. Extended abstract: Incremental knowledge acquisition with selective active learning. In *International Conference on Autonomous Agents and Multiagent Systems*, Istanbul, Turkey, May 4-8, 2015.
6. H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.
7. S. Rosenthal, A. K. Dey, and M. Veloso. How robots' questions affect the accuracy of the human responses. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 1137–1142, Toyama, Japan, September 27-October 2, 2009.
8. K. Salmani and M. Sridharan. Multi-instance active learning with online labeling for object recognition. In *27th International Conference of the Florida AI Research Society*, Pensacola Beach, USA, May 21-23, 2014.
9. B. Settles. *Active Learning*. Morgan & Claypool publishers, 2012.
10. B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, pages 1289–1296. Vancouver, Canada, December 8-11, 2008.
11. R. Swaminathan and M. Sridharan. Towards robust human-robot interaction using multimodal cues. In *Human-Agent-Robot Teamwork Workshop at the International Conference on Human-Robot Interaction*, March 2012.

12. S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM International Conference on Multimedia*, pages 107–118, Ottawa, Canada, September 30-October 5, 2001.
13. C. Zhang and T. Chen. An active learning framework for content-based information retrieval. Technical report, Carnegie Mellon Unversity, Pittsburgh, PA, U.S.A, 2002.