Scaling Up Category Learning for Language Acquisition in Human-Robot Interaction

Luís Seabra Lopes[1,2] and Aneesh Chauhan[2]

[1]Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Portugal
[2]Actividade Transversal em Robótica Inteligente, IEETA, Universidade de Aveiro, Portugal

{ lsl, aneesh.chauhan } @ ua.pt

Abstract

Motivated by the need to support language-based communication between robots and their human users, as well as grounded symbolic reasoning, this paper presents a learning architecture that can be used by robotic agents for long-term and open-ended category acquisition. In this learning architecture, multiple object representations and multiple classifiers and classifier combinations are used. All learning computations are carried out during the normal execution of the agent, which allows continuous monitoring of the performance of the different classifiers. The measured classification successes of the individual classifiers support an attentional selection mechanism, through which classifier combinations are dynamically reconfigured and a specific classifier is chosen to predict the category of a new unseen object. In the current implementation of this learning architecture, base classifiers follow a memory-based approach, in which misclassified instances are simply added to the instance database. The main similarity measures used in the implementation are based on Euclidean distance and on a multi-resolution matching algorithm. Classifier combinations are based on majority voting and the Dempster-Shafer evidence theory. A simple agent, incorporating these learning capabilities, is used to test the approach. A long-term experiment was carried out having in mind the open-ended nature of category learning. With the help of a human mediator, the agent incrementally learned 68 categories of real world objects visually perceivable through an inexpensive camera.

Introduction

Human-robot interaction is currently a very active research field (Fong et al. 2003). The role of social interaction in machine learning and, particularly, in robot learning is being increasingly investigated (Seabra Lopes and Connell 2001; Thomaz and Breazeal, 2006).

Robots are expected to adapt to the non-expert user. This adaptation includes the capacity to take a high-level description of the assigned task and carry out the necessary reasoning steps to determine exactly what must be done. Adapting to the user also implies using the communication modalities of the human user. Spoken language is probably the most powerful communication modality. It can reduce the problem of assigning a task to the robot to a simple sentence, and it can also play a major role in teaching the robot new facts and behaviors. There is, therefore, a trend to develop robots with spoken language capabilities (Seabra Lopes and Connell 2001; Steels and Kaplan 2002; Fong et al. 2003; Seabra Lopes et al. 2005).

Language processing, like reasoning capabilities, involves the manipulation of symbols. By symbol it is meant a pattern that represents some entity in the world by association, resemblance or convention (Seabra Lopes and Chauhan 2007). Association and resemblance arise from perceptual, sensorimotor and functional aspects while convention is socially or culturally established. In classical artificial intelligence, symbolic representations were amodal in the sense that they had no obvious correspondence or resemblance to their referents Harnad (1990) proposed a hybrid approach to the "symbol grounding problem," which consisted of grounding bottom-up symbolic representations in iconic representations and categorical representations.

A distributed view on language origins, evolution and acquisition is emerging in linguistics. This trend emphasizes that language is a cultural product, perpetually open-ended and incomplete, ambiguous to some extent and, therefore, not a code (Love 2004). The study of language origins and evolution has been performed using multi-robot models, with the Talking Heads experiments as a notable example (Steels 1999; Steels 2001). In this case language is transmitted horizontally in the population of robots. Meanwhile, processes where language is vertically transmitted are of particular relevance to robotics applications. In vertical transmission, an agent or population of agents inherits most of its linguistic behavior from a previous generation, or from an independent population (Steels 2003; Steels and Kaplan 2002). Given that language acquisition and evolution, both in human and artificial agents, involve not only internal, but also cultural, social and affective processes, the underlying mechanism has been called "external symbol grounding" (Cowley 2007).

Having in mind the need to support symbolic reasoning and communication mechanisms in artificial agents, this paper investigates how category learning can be implemented in such agents. As in other works reported in the literature, this topic will be explored in a visual category learning domain. Such popular choice is justified by analogies with child development. In fact, in the earliest stages of child language development, most of the vocabulary consists of common nouns that name

concrete objects in the child's environment, such as food, toys and clothes. Gillette et al. (1999) show that, the more imageable or concrete the referent of a word is, the easier it is to learn. So concrete nouns are easier to learn than most verbs, but "observable" verbs can be easier to learn than abstract nouns.

Cognitive models and robotic prototypes have been developed for the acquisition of a series of words or labels for naming certain categories of objects. In general, the success of language acquisition in robots depends on a number of factors (Seabra Lopes and Chauhan, 2007): sensors; active sensing; physical interaction with objects; consideration of the affordances of objects; interaction with the human user; object and category representations; category learning; category membership evaluation. Most of these issues still need to be suitably addressed by robotics researchers.

Roy and Pentland (2002) present a system that learns to segment words out of continuous speech from a caregiver while associating these words with co-occurring visual categories. The implementation assumes that caregivers tend to repeat words referring to salient objects in the environment. Therefore, the system searches for recurring words in similar visual contexts. Word meanings for seven object classes were learned (e.g., a few toy animals, a ball). Steels and Kaplan (2002) use the notion of "language game" to develop a social learning framework through which an AIBO robot can learn its first words with human mediation. The mediator, as a teacher, points to objects and provides their names. Names were learned for three objects: "Poo-Chi," "Red Ball" and "Smiley." The authors emphasize that social interaction must be used to help the learner focus on what needs to be learned. Yu (2005) studies, through a computational model, the interaction between lexical acquisition and object categorization. In a pre-linguistic phase, shape, color and texture information from vision is used to ground word meanings. In a later phase, linguistic labels are used as an additional teaching signal that enhances object categorization. A total of 12 object categories (pictures of animals in a book for small children) were learned in experiments.

The authors of the present paper have previously developed a vocabulary acquisition and category learning system that integrates the user as instructor (Seabra Lopes and Chauhan 2006; Seabra Lopes and Chauhan 2007). The user can provide the names of objects as well as corrective feedback. An evaluation methodology, devised having in mind the open-ended nature of word learning, was proposed and used. On independent experiments, the system was able to learn 6 to 12 categories of regular office objects, associating them to natural language words. Like us, Lovett et al. (2007) also advocate that the key to recognition in the absence of domain expectations (i.e. in open-ended domains) is efficient on-line learning, but the work they describe is still based on the traditional procedures of gathering instances manually, training a recognizer on some of them and finally testing on unseen instances. The most notable feature of this work is the use of qualitative image representations and a specific similarity assessment method. The approach is demonstrated by learning 8 categories of user-drawn sketches. Another recent work also explores continuous learning for visual concepts (Skocaj et al, 2007). They use very simple objects to teach four colour categories (red, green, blue, yellow), two size categories (small, large) and four shape categories (square, circular, triangular, rectangular).

Current approaches to the problem, although quite different from each other, all seem to be limited in the number of categories that can be learned (usually not more than 12 categories). This limitation seems also to affect incremental/lifelong learning systems, not specifically developed for word learning or symbol grounding, such as Learn++ (Polikar, Udpa, Udpa & Honavar 2001) and EBNN (Thrun 1996). Several authors have pointed out the need for scaling up the number of acquired categories in language acquisition and symbol grounding systems (Cangelosi and Harnad 2000; Steels and Kaplan 2002).

Within the field of computer vision, there is recent progress towards systems able to learn larger numbers of categories. The main works are being evaluated on Caltech-101, a well-known database composed of 8677 images of objects of 101 different categories. Recognition accuracy achieved on this problem using 15 training images per category is between 50% and 60% (Grauman and Darrell 2007). However, all works based on the Caltech-101 data follow a traditional train and test approach, rather than focusing on interactive agents with on-line learning capabilities.

In this paper, we present a learning architecture that can be used by robotic agents for long-term and open-ended category acquisition. In this learning architecture, multiple object/category representations and multiple classifiers and classifier combinations are used. All learning computations are carried out during the normal execution of the agent, which allows continuous monitoring of the performance of the different classifiers. The measured classification successes of the base classifiers are used to dynamically reconfigure some of the classifier combinations as well as to select the classifier that will be used to predict the category of a new unseen object.

Agent Architecture

The developed agent is a computer with an attached camera running appropriate perceptual, learning and interaction procedures. The agent's world includes a user, a visually observable area and real-world objects whose names the user may wish to teach. The user, who is typically not visible to the agent, will therefore act as instructor. The user can change de content of the scene, by adding or removing objects.

Using a simple interface, the user can select (by mouse-clicking) any object from the visible scene, thereby enabling shared attention. Then, the user can perform the following teaching actions:

— Teach the object's category name
— Ask the category name of the object, which the agent will predict based on previously learned knowledge
— If the category predicted in the previous case is wrong, the user can send a correction.

Proc. LangRo'2007
978-972-96895-2-9

Main blocks

The agent architecture (figure 1) consists of a perception module, an internal lifelong category learning and recognition module and an action module. The current action system abilities are limited to reporting the classification results back to the user.

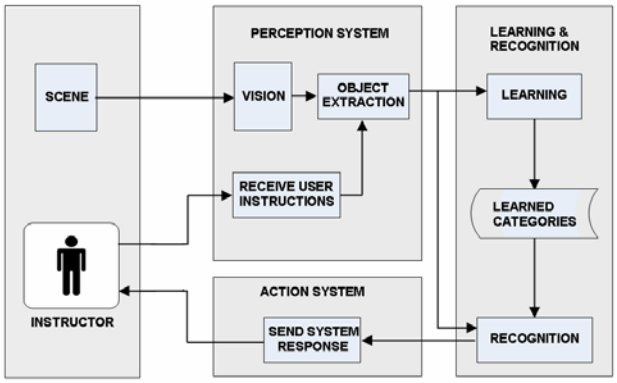

Figure 1 – Agent architecture

The tasks of the perception system include receiving user instructions, capturing images from the camera and extracting object features from images (figure 1). When the user points the mouse to an object in the scene image, an edge-based counterpart of the whole image is generated. The implementation of the canny algorithm, from the publicly available openCV library of vision routines, is used for edge detection. From this edges image, the boundary of the object is extracted taking into account the user pointed position. This is performed using a region growing algorithm and currently assumes that objects don't overlap (or occlude each other) in the image.

Given the boundary of the object, an edges-based counterpart of the object image is extracted from the full scene image (see example in Figure 2).

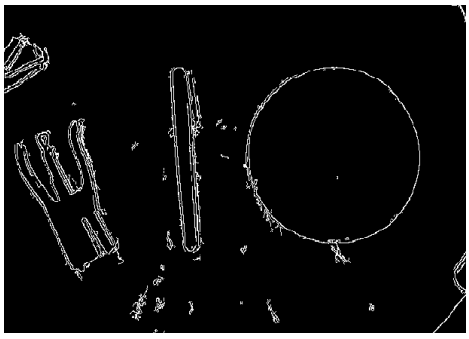

Figure 2 – Edges-based counterpart of a visual scene with three objects

Most of the features used by the classifiers described later in this paper are shape features extracted from this edges image. Only one classifier uses color features. In this case, the original object image is converted to the HSV color format. The primary purpose of using HSV format lies in the fact that most of the color information is in the H component (hue specifies the dominant wavelength of the color in most of its range of values), thus facilitating image analysis based on a single dimension.

The communication between the agent and the human instructor is supported by the perception and action systems. At present, the communication capabilities of the robot are limited to reading the teaching options (teach, ask, correct) in a menu-based interface and displaying classification results. In the future, simple spoken language communication will be supported.

Learning architecture

Language acquisition is highly dependent on the representations and methods used for category learning and recognition. Learning a human language will require the participation of the human user as teacher or mediator (Steels and Kaplan 2002; Seabra Lopes and Chauhan, 2007). A learning system in a robot should support long-term learning and adaptation. Such a system should support supervised, incremental, on-line, opportunistic and concurrent learning and should also be able to improve or optimize its performance through meta-learning (Seabra Lopes and Wang 2002; Seabra Lopes and Chauhan 2007).

The learning architecture proposed here (see Figure 2) was designed to satisfy these requirements. By organizing its categories and instances according to user's feedback, it behaves in a supervised way. It is on-line because it is integrated in the normal activity of the agent. It is incremental and opportunistic because it is able to adjust categories when new instances are observed, rather than requiring that training instances are given in a training phase or according to a pre-defined training schedule. It doesn't involve heavy computations, which facilitates the concurrent handling of multiple learning problems.

This learning architecture is based on the idea that using multiple representations, multiple classifiers and multiple classifier combinations, all potentially complementary of each other, can enhance global performance. Some of these ideas, particularly the use of classifier combinations, are not new in the machine learning literature (Xu et al. 1992). The main innovation in this architecture is that those complementarities are explored in an on-line learning architecture, and a simple form of meta-learning takes advantage of the on-line nature of the learning process, to improve global performance. Teaching and corrective feedback from the human mediator are used to monitor the classification success of the individual classifiers. The measured classification successes of the individual classifiers are used to dynamically reconfigure some of the classifier combinations as well as to select the classifier that will be used to predict the category of a new unseen object.

Feature Spaces

Objects should be described to the learning and classifications algorithms in terms of a small set of informative features. A small number of features will shorten the running time for the learning algorithm.

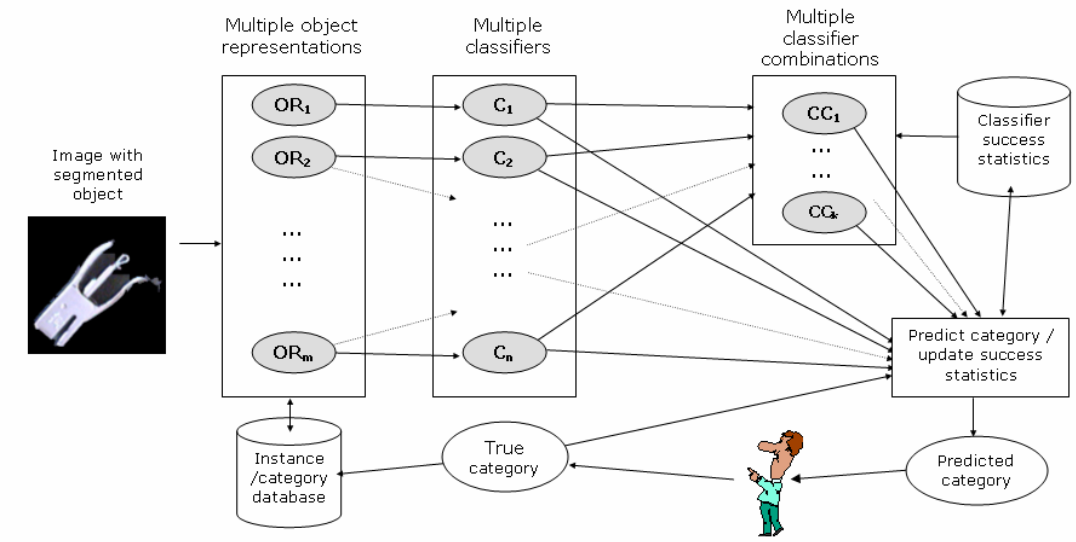

Figure 3 – Learning architecture

Information content of the features will strongly influence the learning performance.

In the approach of this paper, multiple, possibly complementary feature spaces are concurrently explored. Most of these feature spaces result of segmenting the smallest circle enclosing the edges image of the object and centred in its geometric centre. For different feature spaces, such circle is segmented either into a number of slices (Figure 4, left) or a number of concentric layers (Figure 4, right). Current implementation uses 40 slices and 160 layers. Feature spaces based on this kind of segmentation are aimed at capturing shape information. In the following, feature spaces are briefly described.

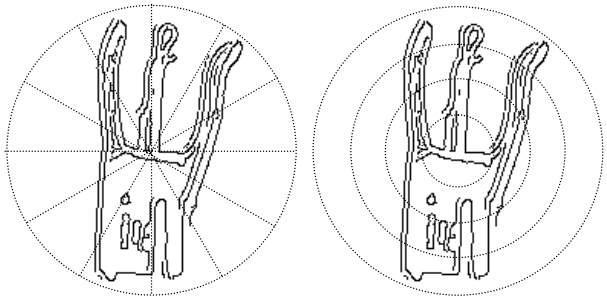

Figure 4 – Segmentation of edges image of an object into slices (left) and layers (right)

Shape slices histogram (SSH). The histogram contains, for each slice, the percentage of edge pixels in that slice with respect to the total number of edge pixels of the object. An example is given in figure 5a for the three objects shown in figure 2. Given the rotation-dependent nature of this feature space, similarity (or distance) between instances must be computed as the maximum similarity (or minimum distance) between the respective feature vectors as they are circularly rotated relative to each other.

Area (AREA). This feature space is composed of a single feature, *area*, defined as the total number of pixels of the object. This is the only scale-dependant feature space used in this work.

Shape slices normalized radii averages (SSNRA). For each slice, i, the average radius of all pixels in that slice, R_i, is computed. In this feature space, an object represented by a vector $\bar{r} = r_i \,..\, r_{40}$, where $r_i = R_i / R$ and R is the average of all R_i. This is the core of the feature space used in previous work (Seabra Lopes and Chauhan, 2007). An example is given in figure 5b for the three objects shown in figure 2. As in shape slices histogram, similarity computations involve rotations.

Normalized radius standard deviation (RADSD). This is another feature space composed of a single feature. Its value is the standard deviation of the normalized radii averages, $r_i \,..\, r_{40}$, mentioned in the previous paragraph.

Shape slices normalized radii standard deviations (SSNRSD). For each slice, i, the radius standard deviation of all pixels in that slice, S_i, is computed. In this feature space, an object is represented by a vector $\bar{s} = s_i \,..\, s_{40}$, where $s_i = S_i / R$ and R is the average radius as mentioned above. An example is given in figure 5c for the three objects shown in figure 2. As in other representations based on shape slices, similarity computations involve rotations.

Shape layers histogram (SLH). The histogram contains, for each layer, the percentage of edge pixels with respect to the total number of edge pixels of the object. This feature space is not only scale-invariant, but also rotation-invariant. An example is given in figure 5d for the three objects shown in figure 2.

Color ranges (COLOR). In this feature space, an object is represented by a set of the main colors of the object. Each color is represented as a range of hue values in HSV color space. These color ranges are

Proc. LangRo'2007
978-972-96895-2-9

extracted from a color histogram using a simple method presented in a previous paper (Seabra Lopes et al 2007).

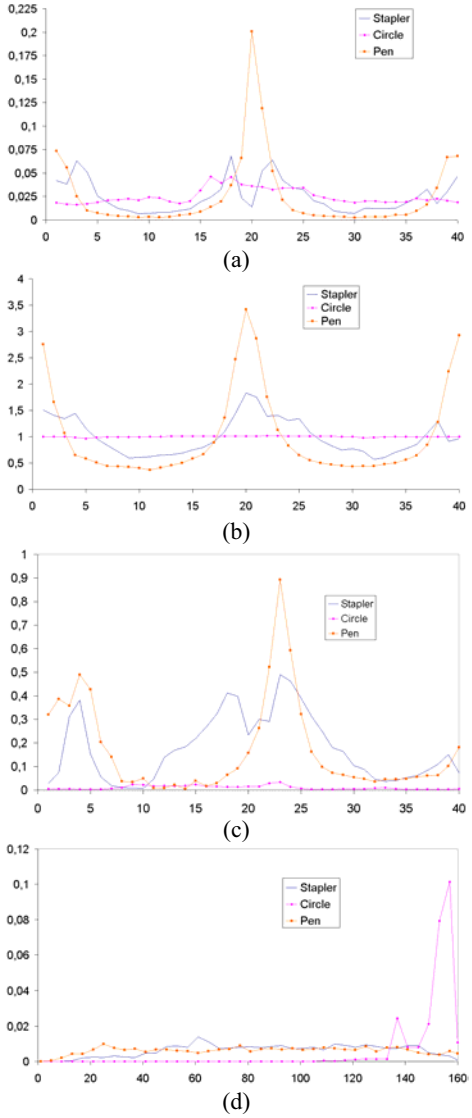

Figure 5 – Different types of shape features for the three objects in figure 2: (a) Shape slices histogram; (b) Shape slices normalized radii averages; (c) Shapes slices normalized radii standard deviations; (d) Shape layers histogram.

Similarity and Membership Measures

Categorization of a new previously unseen instance involves ranking the known categories according to measures of membership of the instance to each of the categories. In turn, computing membership measures often involves evaluating similarities and/or distances between instances.

Euclidean measures

Since objects are represented as feature vectors in most of the features spaces described above, an obvious similarity measure is inverse Euclidean distance. For two objects $\bar{x}$ and $\bar{y}$, with distance $D = \|\bar{x} - \bar{y}\|$, inverse Euclidean distance is given by $1/D$.

In instance-based classifiers, membership to a category is evaluated by computing and combining the similarities of the target object to the known instances of the category. In the present work, assuming that categories are homogeneous, i.e. that there are no significant intra-class variations, averaging the similarities of the target object to the known instances of the category seems an appropriate strategy for computing membership measures.

One of the membership measures used in this work is, therefore, computed by inverting and normalizing the average Euclidean distance of the target object to the instances of the given category C_i, as follows:

$$EuclidMem(C_i) = \frac{N}{D_i \sum_{k=1}^{N} (1/D_k)},$$

where N is the number of categories, i, k=1, …, N, and D_i and D_k are the average Euclidean distances of the target object to the known instances of categories C_i and C_k, respectively. The membership values $EuclidMem(C_i)$ sum to 1.0, allowing their use as evidence in Dempster-Shafer combinations.

Multi-resolution measures

In the present work, similarity is alternatively measured through a multi-resolution matching algorithm similar to the matching algorithm used in the recently proposed *pyramid match kernel* (Grauman and Darrell 2007). This kernel function was designed to enable the application of kernel-based learning methods to domains where objects are represented by unordered and variable-sized sets of features, such as sets of local features in computer vision. In this kernel, each feature set is mapped to a histogram pyramid, i.e. a multi-resolution histogram preserving the individual features distinctness at the base level. Then, the histogram pyramids are matched using a weighted histogram intersection computation.

The feature spaces used in the present work, as described above, are ordered and have a constant dimension, so mapping these representations to multi-resolution pyramids is direct. Then, the same basic matching algorithm can be applied.

The pyramid match score for two objects $\bar{x}$ and $\bar{y}$ is given by:

$$P_\Delta(\bar{x}, \bar{y}) = \sum_{i=0}^{L-1} w_i N_i(\bar{x}, \bar{y})$$

where L is the number of pyramid layers, $w_i = 2^i$ is the weight of layer i and N_i measures the additional matching at layer i, as given by:

$$N_i(\bar{x}, \bar{y}) = I(F_i(\bar{x}), F_i(\bar{y})) - I(F_{i-1}(\bar{x}), F_{i-1}(\bar{y}))$$

where $F_i(\bar{x})$ is the feature representation of object $\bar{x}$ at layer i and $I()$ is an intersection function which measures the overlap of two objects as follows:

$$I(A, B) = \sum_j \min(A_i, B_i)$$

Note that this type of matching applies not only to histograms, as done by Grauman and Darrell, but also to other feature vectors that are normalized by some constant R, as it happens in the "shape slices normalized radii averages" feature space described above. The use of pyramid matching in the present work extends a previous, simpler idea of the authors, which consisted of including in feature spaces block averages computed based on base-level ordered feature vectors (Seabra Lopes and Chauhan 2007; Seabra Lopes and Camarinha-Matos 1998).

Based on the pyramid match score, the following category membership measure for a particular target object and category C_i can be computed:

$$PyramidMem(C_i) = \frac{N \cdot P_i}{\sum_k P_k}$$

where N is the number of categories, i, k=1, …, N, and P_i and P_k are the average pyramid match scores of the target object to the known instances of categories C_i and C_k, respectively. The $PyramidMem(C_i)$ membership values sum to 1.0, allowing their use as evidence in Dempster-Shafer combinations.

Categorization

In the present work, categories are simply represented by sets of known instances. The known instances that are stored are those explicitly taught by the human user and also those objects that the agent failed to categorize correctly, leading to corrective feedback from the user. The agent, therefore, doesn't add to its instance database those objects that it was able to categorize correctly.

Categorizing a new previously unseen object involves computing measures of membership of the object to the known categories. The category with highest membership measure for the target object is returned. These computations are carried out by classifiers. In the present work, multiple classifiers and multiple classifier combinations are used.

Base classifiers

The use of a specific membership measure with a specific feature space results in a specific "base classifier". The following base classifiers were included in the implementation:

- Classifiers using single-dimension feature spaces with Euclidean membership measurement: "Area" (AREA); "Normalized radius standard deviation" (RADSD).
- Classifiers using feature vectors with Euclidean membership measurement: "shape slices histogram" (SSH-EM), "shape slices normalized radii averages" (SSNRA-EM), "shape slices normalized radii standard deviations" (SSNRSD-EM) and "shape layers histogram" (SLH-EM).
- Classifiers using feature vectors (the same as in the previous group) with pyramid membership measurement: SSH-PM, SSNRA-PM, SSNRSD-PM and SLH-PM.
- Classifier based on a color-based category representation and membership measure (COLOR) presented elsewhere (Seabra Lopes et al. 2007).

In total, therefore, the implementation includes 11 base classifiers.

Classifier combinations and meta-learning

The complete learning and categorization approach includes classifier combinations. Some of the classifier combinations are dynamically reconfigured according to the observed success of the base classifiers. Classification success rates computed over the last N iterations (N=50 was used) is the performance measure used to guide reconfiguration of classifier combinations. This introduces a meta-learning component in the category learning system.

Dempster-Shafer combinations

The Dempster-Shafer theory of evidence is a powerful tool for representing and combining uncertain knowledge (Shafer 1976). It is based on a basic belief assignment, i.e. a mass function $m(A)$ that assigns a value in [0,1] to every subset A of a set of mutually exclusive propositions θ. The belief in the composite proposition $B \subseteq \theta$ is given by the sum of $m(A)$ for all $A \subseteq B$. The belief in θ sums to 1.0. In this theory, when multiple evidences allow to derive multiple basic belief assigments, these evidences can be combined. In particular, two basic belief assignments m_1 and m_2 can be combined by the following rule:

$$m(C) = \frac{\sum_{A,B,A \cap B=C} m_1(A) \cdot m_2(B)}{1 - \sum_{A,B,A \cap B=\emptyset} m_1(A) \cdot m_2(B)}$$

This rule is the basis of a well known method for combining multiple classifiers (Xu et al. 1992; Al-Ani and Deriche, 2002). Each classifier provides evidence that is expressed as a basic probability assignment. In the work of this paper, the membership measures described above (Euclidean-based and pyramid-based) are directly used as masses. As mentioned before, these membership measures are normalized to sum to 1.0.

Sets containing more than one category are assigned a mass of 0.0, so the approach comes close to the Bayesian combination approach. The main difference is that normalized membership measures are used instead of conditional probabilities. These conditional probabilities could be estimated based on the confusion matrixes of each classifier. The classical way of doing this is to acquire a confusion matrix for each classifier in a preliminary training/testing phase. This approach, however, is not viable in a long-term / open-ended learning scenario. Is such a scenario, therefore, the alternative would be to build the confusion matrixes on-line. This would imply that, in an initial stage as well as after the introduction of a new category, the conditional probabilities would be heavily biased by the specific cases seen so far. We did some exploratory experiments in this direction and observed that classifier combinations based on conditional probabilities start

behaving poorly, but eventually catch up with classifier combinations based on membership measures. However, even in the long run, conditional probabilities didn't seem to be able to significantly outperform membership measures, as far as classifier combinations are concerned.

Four Dempster-Shafer classifier combinations were included in the implementation, namely combinations of the top two, three, four and five most successful classifiers (respectively DS2TOP, DS3TOP, DS4TOP and DS5TOP). Since the classification success of each classifier is re-evaluated in each teaching/learning interaction with the human user, these classifier combinations are also dynamically reconfigured in each such opportunity.

Majority voting combinations

Voting methods are also well known in classifier combinations (Xu et al, 1992; Kittler et al, 1998). In the implementation, two dynamically reconfigured classifier combinations based on majority voting were included: majority voting of the top three and five most successful classifiers (respectively MAJ3TOP and MAJ5TOP). In addition, a classifier combination based on majority voting of all previously described classifiers (MAJORITY-ALL) was also included.

The Predicted Category

The internal computations described until now culminate in a category prediction that is communicated to the interlocutor(s) of the agent, typically a human user. This category will be the category predicted by the currently most successful classifier, considering all base classifiers and classifier combinations described above.

Experimental Evaluation Protocol

The word/category learning literature has some common features. One of them is the limitation on the number of learned words. The known approaches have been demonstrated to learn up to 12 words.

The other common feature is the fact that the number of words is pre-defined. This is contrary to the open-ended nature of the word learning domain. Then, given that the number of categories is pre-defined, the evaluation methodology usually consists of extracting certain measures on the learning process (Roy and Pentland 2002; Steels and Kaplan 2002; Yu 2005; Skocaj et al. 2007; Lovett et al, 2007). Some authors plot this type of measures versus training time or number of examples. As the number of words/categories is pre-defined, the plots usually show a gradual increase of these measures and the convergence to a "final" value that the authors consider acceptable.

However, robots and software agents are limited in their perceptual abilities and, therefore, cannot learn arbitrarily large numbers of categories, particularly when perception does not enable the detection of small between-category differences. As the number of categories grows, learning performance will evolve, with phases of performance degradation followed by recovery, but will eventually reach a breakpoint.

A well-defined teaching protocol can facilitate the comparison of different approaches as well as the assessment of future improvements. With that in mind, the teaching protocol of figure 6 was previously proposed (Seabra Lopes and Chauhan 2007). For clarity, its presentation is repeated here.

```
introduce Class0;
n = 1;
repeat {
   introduce Classn;
   k = 0;
   repeat {
      Evaluate and correct classifiers;
      k ← k + 1;
   } until ( ( average precision >
                precision threshold and k≥n)
             or
            (user sees no improvement in precision));
   n ← n + 1;
} until (user sees no improvement in precision).
```

Figure 6 – Experimental evaluation protocol

This protocol is applicable for any open-ended class learning domain. For every new class the instructor introduces, the average precision of the whole system is calculated by performing classification on all classes for which data descriptions have already been learned. Average precision is calculated over the last $3\times n$ classification results (n being the number of classes that have already been introduced). The precision of a single classification is either 1 (correct class) or 0 (wrong class). When the number of classification results since the last time a new class was introduced, k, is greater or equal to n, but less than 3×n, the average of all results is used. The criterion that indicates that the system is ready to accept a new object class is based on the precision threshold.

Experimental Results

Experiments were conducted according to this protocol. The set of categories and the set of training instances were not established in advance. As categories were learned, new objects were fetched from the surrounding office environment and used to introduce new categories. Many objects were brought from the homes of the authors for proceeding with the experiments until the breakpoint was reached.

The experiments lasted for several days and a total of 3767 question/correction iterations (figure 7). In total, it was possible to teach 68 categories of real-world objects, which can be roughly grouped as follows: office objects – 40%; child toys – 20%; other home objects – 20%; other – 20%. Figure 8 displays one sample image per category. During the

teaching/learning process, the agent stored a total of 1168 training instances.

Figure 7 displays the evolution of classification precision versus number of question/correction iterations. As observed in previous work (Seabra Lopes and Chauhan 2007; Seabra Lopes et al. 2007), classification precision degrades after the introduction of each new category, then eventually recovers.

Towards the limit of the category discrimination abilities of the agent, learning starts to take longer. From figure 7, we see that most categories were learned in the first ~2000 iterations (exactly 60 categories), while in the remaining ~1800 iterations it was possible to learn only 8 additional categories. The breakpoint is also clearly visible in figure 9 which displays the evolution of the average number of training instances per category versus the increasing number of categories. After learning the first 30 categories, the system had stored less than 4 instances per category. While learning additional 30 categories, the number of instances per category continued to grow according to a linear trend, reaching an average close to 9 instances per category. Finally, while learning the last 8 categories, the number of instances per category abandoned the linear evolution trend and jumped to 17.

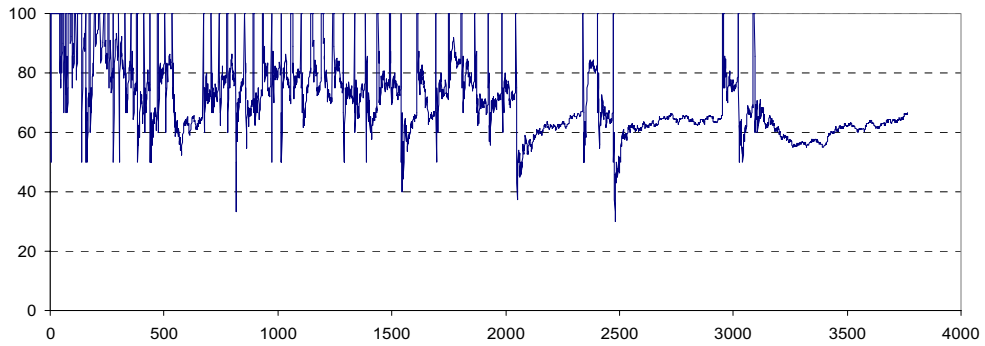

Figure 7 – Evolution of classification precision versus number of question/correction iterations

Figure 8 – Sample images of all acquired categories

Proc. LangRo'2007
978-972-96895-2-9

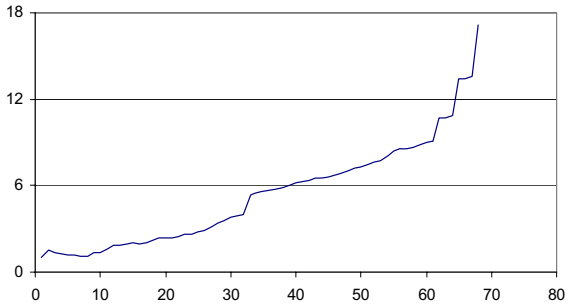

Figure 9 – Evolution of number of training instances per category versus number of acquired categories

Table I – Classification success rates for all classifiers

Single dimension classifiers	
AREA	44.6
RADSD	27.4
Shape feature vectors with Euclidean membership	
SSH-EM	2.8
SSNRA-EM	2.9
SSNRSD-EM	2.0
SLH-EM	30.8
Shape feature vectors with pyramid membership	
SSH-PM	46.2
SSNRA-PM	64.6
SSNRSD-PM	8.2
SLH-PM	43.1
Color-based classifier	
COLOR	8.7
Dempster-Shafer combinations	
DS2TOP	57.3
DS3TOP	57.5
DS4TOP	56.9
DS5TOP	64.2
Majority voting combinations	
MAJ3TOP	65.3
MAJ5TOP	63.9
Majority voting of all other classifiers	
MAJORY-ALL	70.6

The analysis of the performance of the individual classifiers is also relevant (Table I). Classifiers based on Euclidean membership perform very poorly. The best classifier in this group was SLH-EM (shape layers histogram with Euclidean membership) with an average precision of 30%. One of the single dimension classifiers performed better than that (AREA, 45%). Classifiers based on pyramid membership measurement performed far better than the Euclidean ones (e.g. SSNRA-PM, 65%).

Dynamically reconfigured Dempster-Shaffer combinations were in the range of 57% to 64%. Dynamically reconfigured majority voting combinations were in the range of 64% to 65%. Finally, the majority voting of all other classifiers (MAJORITY ALL) achieved a precision of 70.6%.

The externally observable performance of the agent was very close to MAJORITY-ALL, exactly 70.0%. Note that, as mentioned before, the predictions of the agent are those of the current most successful classifier. In 64% of experiment time, MAJORITY-ALL was the best classifier. Other classifiers were the most successful for shorter amounts of time: MAJ3TOP with a share of 10%, SSNRA-PM with a share of 8% and DS5TOP with a share of 4%.

Conclusions

This paper presented a category learning architecture with several innovations. One of them is the use of multiple representations, multiple classifiers and multiple classifier combinations, all potentially complementary of each other. Although common in off-line learning, this approach has not been explored for on-line learning methods. Another innovation is the use of an attentional selection mechanism to reconfigure classifier combinations as well as to select the classifier that is used in a specific situation. Although our goal is not to emulate human category learning, some parallels can be drawn with research in that field (Ashby and Obrien 2005; Kruschke 2005). In particular, researchers have been recently moving toward the conclusion that human category learning relies on multiple memory systems and multiple representations. The also recently emphasized role of attentional selection, i.e. a mechanism of focusing on specific features or representations based on recent experience, can be paralleled with our mechanism of dynamically selecting and reconfiguring classifiers.

The shape-based feature spaces are also an original proposal of the authors in an attempt to develop computationally light classifiers that can be used in an on-line classifier combination architecture. The application of the pyramid matching algorithm of (Grauman and Darrell, 2007) to feature spaces where objects described, not by histograms, but by other normalized feature vectors is also a contribution of this paper, which actually produced excellent results. The extreme differences in performance between SSNRA EM (2.9%) and SSNRA-PM (65%) illustrate this point.

Overall, our approach seems to outperform several previous works initially cited. While previous approaches enabled learning of up to 12 categories, the proposed approach enabled learning of 68 categories in a long-duration experiment.

Besides the overall success of this work, compared to previous works with similar goals, the results provide support to some of the “ingredients” of the approach. In particular, the use of pyramid matching proved far more effective than Euclidean distance in similarity assessment. Also, majority voting proved successful in maximizing overall performance. The results don’t provide irrefutable evidence in favor of the proposed attentional selection mechanism. Actually, selecting the “current best” classifier led the agent to perform slightly worse (70%) than the majority voting classifier (70.6%). Future experiments will be designed to enable drawing conclusive results concerning attentional selection.

Aknowledgements

The Portuguese Research Foundation (FCT) supported this work under contract POSI/SRI/48794/2002 (project "LANGG: Language Grounding for Human-Robot Communication"), which is partially funded by FEDER.

References

Al-Ani, A. and Deriche, M., (2002) A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence, *Journal of Artificial Intelligence Research*, 17, 333-361.

Ashby, F.G. and J.B. Obrien (2005) Category Learning and Multiple Memory Systems, *Trends in Cognitive Science*, 9(2), 83-89.

Barsalou, L. (1999). Perceptual symbol systems, *Behavioral and Brain Sciences*, 22(4), 577-609.

Cangelosi, A. & Harnad, S. (2000). The adaptive advantage of symbolic theft over sensorimotor toil: Grounding language in perceptual categories. *Evolution of Communication*, 4(1), 117-142.

Cowley, S. J. (2007). Distributed language: Biomechanics, functions and the origins of talk. In C. Lyon, C. Nehaniv & A. Cangelosi (Eds.), *Emergence of communication and language*. Springer, 105-127.

Fong, T., Nourbakhsh, I. & Dautenhahn, K. (2003). A survey of socially interactive robots: Concepts, design, and applications, *Robotics and Autonomous Systems*, 42, 143-166.

Gillette, J. Gleitman, H., Gleitman, L. & Lederer, A. (1999). Human simulations of vocabulary learning, *Cognition*, 73, 135-176.

Grauman, K. and T. Darrell (2007) The Pyramid Match Kernel: Efficient Learning with Sets of Features. *Journal of Machine Learning Research*, 8, 725-760.

Harnad, S. (1990). The symbol grounding problem, *Physica D*, 42, 335-346.

Kitler, J., R.P.W. Duin and J. Matas (1998) On Combining Classifiers, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(3), 226-239.

Kruschke, J. K. (2005). Category Learning. In: K. Lamberts and R. L. Goldstone (Eds.), *The Handbook of Cognition*, Chapter 7, 183-201.

Love, N. (2004). Cognition and the language myth, *Language Sciences*, 26, 525-544.

Lovett, A., M. Dehghani and K. Forbus (2007) Incremental Learning of Perceptual Categories for Open-Domain Sketch Recognition, *Proc. Int. J.Conf. Artificial Intellig. (IJCAI'07)*, p. 447-452.

Polikar, R., Udpa, L., Udpa, S. S. & Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *31*(4), 497-508.

Roy, D., and A. Pentland (2002). Learning words from sights and sounds: A computational model, *Cognitive Science*, 26, 113-146.

Seabra Lopes, L. and L.M. Camarinha-Matos (1998) Feature Transformation Strategies for a Robot Learning Problem, *Feature Extraction, Construction and Selection. A Data Mining Perspective*, H. Liu and H. Motoda (edrs.), Kluwer Academic Publishers.

Seabra Lopes, L. & Chauhan, A. (2006) One-Class Lifelong Learning Approach to Grounding, *Workshop on External Symbol Grounding. Book of Abstracts and Papers*, Plymouth, UK, 15-23.

Seabra Lopes, L. and A. Chauhan (2007) How many Words can my Robot learn? An Approach and Experiments with One-Class Learning, *Interaction Studies*, 8(1), 53-81.

Seabra Lopes, L., A. Chauhan and J. Silva (2007) Towards long-term visual learning of object categories in human-robot interaction, *Proc. 2nd Intelligent Workshop on Intelligent Robotics - 13th Portuguese Conference on Artificial Intelligence (EPIA2007)*, to appear.

Seabra Lopes, L. and J.H. Connell (2001) Semisentient Robots: Routes to Integrated Intelligence, *IEEE Intelligent Systems*, vol. 16(5), p. 10-14.

Seabra Lopes, L., Teixeira, A. J. S., Quinderé, M. & Rodrigues, M. (2005). From robust spoken language understanding to knowledge acquisition and management. *Proc. Interspeech 2005*, Lisbon, Portugal, 3469-3472.

Shafer, G. (1976) *A Mathematical Theory of Evidence*, Princeton University Press.

Skocaj, D., G. Berginc, B. Ridge, A. Stimec and N. Hawes (2007) A System for Continuous Learning of Visual Concepts, *Proc. Int. Conf. on Computer Vision Systems (ICVS2007)*, Bielefeld, Germany.

Steels, L. (2001) Language games for autonomous robots, *IEEE Intelligent Systems*, 16(5), 16-22.

Steels, L. & Kaplan, F. (2002). AIBO's first words: The social learning of language and meaning, *Evolution of Communication*, 4(1), 3-32.

Steels, L. (2003) Evolving Grounded Communication for Robots, *Trends in Cognitive Science*, 7(7), 308-312.

Thomaz, A. L., and C. Breazeal (2006) Transparency and Socially Guided Machine Learningm, *Proc. 5th International Conference on Developmental Learning (ICDL)*.

Thrun, S. (1996). *Explanation-based neural network learning: A lifelong learning approach*. Boston, MA: Kluwer.

Xu, L., Krzyzak, A., Suen, C.Y. (1992) Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Trans. Systems, Man and Cybernetics*, 22 (3), 418-435.

Yu, C. (2005). The emergence of links between lexical acquisition and object categorization: A computational study, *Connection Science*, 17(3-4), 381-397.