

John Benjamins Publishing Company



This is a contribution from *Interaction Studies* 8:1
© 2007. John Benjamins Publishing Company

This electronic file may not be altered in any way.

The author(s) of this article is/are permitted to use this PDF file to generate printed copies to be used by way of offprints, for their personal use only.

Permission is granted by the publishers to post this file on a closed server which is accessible to members (students and staff) only of the author's/s' institute.

For any other use of this material prior written permission should be obtained from the publishers or through the Copyright Clearance Center (for USA: www.copyright.com).

Please contact rights@benjamins.nl or consult our website: www.benjamins.com

Tables of Contents, abstracts and guidelines are available at www.benjamins.com

How many words can my robot learn?

An approach and experiments with one-class learning*

L. Seabra Lopes and A. Chauhan

Universidade de Aveiro, Portugal

This paper addresses word learning for human–robot interaction. The focus is on making a robotic agent aware of its surroundings, by having it learn the names of the objects it can find. The human user, acting as instructor, can help the robotic agent ground the words used to refer to those objects. A lifelong learning system, based on one-class learning, was developed (OCLL). This system is incremental and evolves with the presentation of any new word, which acts as a class to the robot, relying on instructor feedback. A novel experimental evaluation methodology, that takes into account the open-ended nature of word learning, is proposed and applied. This methodology is based on the realization that a robot's vocabulary will be limited by its discriminatory capacity which, in turn, depends on its sensors and perceptual capabilities. The results indicate that the robot's representations are capable of incrementally evolving by correcting class descriptions, based on instructor feedback to classification results. In successive experiments, it was possible for the robot to learn between 6 and 12 names of real-world office objects. Although these results are comparable to those obtained by other authors, there is a need to scale-up. The limitations of the method are discussed and potential directions for improvement are pointed out.

Keywords: human–robot interaction, external symbol grounding, word learning, one-class learning, experimental methodologies

Introduction

The robotics community is increasingly involved in designing and developing user-friendly robots, that is, robots that are flexible, adaptable and easy to command and instruct (Breazeal & Scassellati, 2000; Chatila, 2004; Fong, Nourbakhsh & Dautenhahn, 2003; Seabra Lopes & Connell, 2001b). A user-friendly robot must be prepared to adapt to the 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. Reasoning capabilities such as action sequence planning or logical inference are, by definition, based on manipulating symbolic representations.

User-friendliness also includes understanding and 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 (Levinson, Squire, Lin & McClain, 2005; Seabra Lopes, 2002; Seabra Lopes, Teixeira, Quinderé & Rodrigues, 2005; Steels & Kaplan, 2002; see also several reports in Seabra Lopes & Connell, 2001a).

This paper addresses word learning for human–robot interaction. The learning paradigm of choice for this work is one-class learning. An incremental learning system based on Support Vector Data Description (Tax, 2001) was developed to support the grounding of word meanings. Given the open-ended nature of word learning, this system is designed to support the concurrent/opportunistic learning of an arbitrary number of classes/words. Through mechanisms of shared attention and corrective feedback, a human user, acting as an instructor, can help the robot ground the words used to refer to real-world office objects that it finds in its environment. A novel experimental evaluation methodology is proposed for word learning. This methodology took two main considerations into account. On the one hand, word learning is an open-ended domain. On the other hand, an agent’s vocabulary will be limited by its discriminatory capacity which, in turn, depends on its sensors and perceptual capabilities. This methodology can be useful for comparing the word learning capabilities of different agents and for assessing research progress on scaling-up to larger vocabularies.

Situating the problem

Symbol and language grounding

Both reasoning and language processing involve the manipulation of symbols. By *symbol* we mean a pattern that represents some entity in the world by association, resemblance or convention. Association and resemblance arise from perceptual, sensorimotor and functional aspects while convention is socially or culturally established.

The advent of computers encouraged people to start developing “intelligent” artifacts, including artifacts with human-level intelligence (Turing, 1950). As reasoning and language are key components of intelligence, the first few decades of

research on artificial intelligence (AI) focused on first-order logic, semantic networks, logical inference, search techniques and natural language processing. Symbol systems in AI were theorized by Simon and Newell in successive publications since the 1970s, and became the dominant model of the mind in cognitive science (see the survey and critical analysis of Anderson & Perlis, 2002).

These symbolic representations were amodal in the sense that they had no obvious correspondence or resemblance to their referents (Barsalou, 1999). As aspects related to perception and sensorimotor control were largely overlooked, establishing the connection between symbols and their referents remained an open issue. The problem of making the semantic interpretation of a formal symbol system intrinsic to that system was called “the symbol grounding problem” (Harnad, 1990). Eventually, the limitations of classical symbolic AI led to a vigorous reaction, generally known as “situated” or “embodied” AI, and, in particular, to the “intelligence without representation” views of Brooks (1991).

In the meantime, the resurgence of connectionism led various authors to propose hybrid symbolic/connectionist approaches. In particular, Harnad (1990) proposed a hybrid approach to the “symbol grounding problem,” which consists of grounding bottom-up symbolic representations in iconic representations (sensory projections of objects) and categorical representations (learned or innate connectionist functions capable of extracting invariant features from sensory projections). Elementary symbols are the names of these categories. More complex representations are obtained by aggregating elementary symbols.

The increasing concern with perception and sensorimotor control, both in the AI and robotics communities, was paralleled in cognitive science. Barsalou (1999) develops a theory on “perceptual symbol systems,” which takes up the classical (perceptual) view of cognition. A “perceptual symbol” is viewed as an unconscious neural representation that represents some component of perceptual experience. Related perceptual symbols become organized into a kind of category or concept, called a simulator. The simulator is able to produce limitless simulations (conscious mental images of members of the category) even in the absence of specific perceptual experience. Simulators can be aggregated in frames to produce simulators for more complex categories. Linguistic symbols are viewed as perceptual symbols for spoken or written words. As linguistic simulators develop, they become associated with the simulators of the entities to which they refer.

Taking a broader perspective, Clark (1997) sees control of embodied action as an emergent property of a distributed system composed of brain, body and environment. However, Clark rejects radical anti-representationalist approaches and accepts the need for representations geared to specific sensorimotor needs. He also emphasizes the importance of external scaffolding, that is, the support provided to thought by the environment and by public language.

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, 2001). Steels and Kaplan (2002) have reported a related robotic approach to language acquisition. 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, 2007a).

The symbol grounding problem was originally formulated as a problem of formal symbol systems and classical AI (Harnad, 1990). However, most research on symbol grounding has been taking place within cognitive science, usually with a strong cognitive modeling flavor and, therefore, with concerns for psychological plausibility (Cangelosi, 2005). However, it is becoming necessary to study symbol and language grounding from an engineering perspective, that is, having in mind the development of machines with reasoning and language skills suitable for practical applications. The main criterion here is no longer the psychological plausibility of the approaches but their utility. This is consistent with a modern view of AI, which no longer concentrates on solving problems by simulating human intelligence, but rather on developing practically useful systems with the most suitable approaches.

Word learning

This paper addresses word learning with a motivation coming from the area of human–robot interaction. Word learning is a basic language acquisition task and, therefore, relies on external symbol grounding mechanisms. For artificial agents, the problem is designing suitable mechanisms for this. Cognitive models and robotic prototypes have been developed for the acquisition of a series of words or labels for naming certain categories of objects. The next paragraphs provide an overview of some of the main published models and prototypes.

Harnad, Hanson, and Lubin (1991, 1995) study categorical perception effects (within-category compression and between-category expansion) with a three-layer feed-forward network. The work involved the sorting of lines into three categories (“short,” “middle,” “long”). Plunkett and collaborators (Plunkett, Sinha, Moller & Strandsby, 1992; Plunkett & Sinha, 1992) use a dual-route connectionist architecture with auto-associative learning for studying language production and understanding. Retinal and verbal information were present in both input and output layers, and the network had two hidden layers. After training, the network could be

used both for language generation (object category name, given visual perception) and understanding (object visualization given the name). Sales and Evans (1995) used a dual-route architecture based on “weightless artificial neurons.” They claim that their system can easily acquire 50 grounded nouns, although the demonstration is limited to three object categories (“apple,” “jar” and “cup”).

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. Two-dimensional histograms for multiple views for representing each object and a chi-squared distance metric were used for comparing objects. Word meanings for seven object classes were learned (e.g., a few toy animals, a ball.)

Steels and Kaplan (2002, see also Steels, 2001) 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. The robot uses color histograms and an instance-based learning method to learn word meanings. The mediator can also ask questions and provide feedback on the robot’s answers. Names were learned for three objects: “Poo-Chi,” “Red Ball” and “Smiley.” While Harnad (1990) argued for bottom-up grounding of symbolic representations into categories, Steels and Kaplan show, with concrete robotic experiments, that unsupervised category formation may produce categories that are completely unrelated to the categories that are needed for grounding the words of the used language. They therefore conclude that social interaction must be used to help the learner focus on what needs to be learned. This is in line with previous linguistic and philosophical theories, including the Sapir-Whorf thesis (Talmy, 2000; for a related recent study, see Yoshida & Smith, 2005).

Levinson et al. (2005) describe a robot that learns to associate meanings using a cascade of hidden Markov models. After about 30 minutes of training, the robot is able to associate linguistic expressions with four objects: a green ball, a red ball, a toy dog and a toy cat. The linguistic expressions designate two abstract categories (“animal” and “ball”) and four concrete categories (“green ball,” “red ball,” “dog” and “cat”).

Yu (2005) studies, through a computational model, the interaction between lexical acquisition and object categorization. In a pre-linguistic phase, shape (histograms), color and texture (Gabor filters) information from vision is used to ground word meanings. After the application of PCA, Gaussian mixtures are used to cluster the category description. 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) was used for experiments.

Greco, Riga, and Cangelosi (2003) study grounding transfer, that is, the process of building composed symbolic representations from grounded elementary symbols, as originally proposed by Harnad (1990). They present two simulations based on connectionist architectures: one with four shape categories, four texture categories and four composed categories, and the other with three color categories, three shape categories and nine composed categories.

This survey includes quite different approaches, at all levels (e.g., embodiment, teaching/mediation, complexity of the named objects, feature extraction, learning method, number of training examples, learning time.). Nevertheless, they all seem to be limited in the number of classes or categories that can be learned (this number varies between 3 and 12 in the cited works). This limitation seems also to affect incremental/lifelong learning systems not specifically developed for word learning or symbol grounding. That is the case for Learn++ (Polikar, Udpa, Udpa & Honavar, 2001) and EBNN (Thrun, 1996). Steels and Kaplan (2002) and Cangelosi (2005) have already pointed out the need for scaling up the number of acquired categories for symbol/language grounding.

As can be seen from the survey, researchers have focused on naming visually observable concrete objects. This will also be the focus of the present paper. Interestingly, in the earliest moments 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. The rest includes routine social words, proper nouns, animal sounds, and almost no verbs or function words. The over-representation of common nouns (and corresponding under-representation of verbs) can be observed until the third birthday. Gillette, Gleitman, Gleitman and Lederer (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. In learning words, children show several systematic attentional biases. For concrete solid objects, there is a bias towards generalizing object names to other instances based on shape (Gershoff-Stowe & Smith, 2004; Samuelson & Smith, 2005; Smith & Samuelson, in press). Concerning developmental evolution, vocabulary starts with about 10 words around the age of one, increases to about 300 words in the second year and continues increasing steadily until adolescence (Bates, Thal, Finlay & Clancy, 2002; Crystal, 1987). The average vocabulary for adults is in the order of several tens of thousands. Early categories associated to words are often not consistent with the categories of adults. It has been observed that, in later lexical development (ages of 5 to 14), categories are gradually reorganized to converge to the categories of adults (Ameel, Malt & Storms, 2006).

Learning requirements

Language grounding is highly dependent on the techniques and methods being used for learning. Learning a human language will require the participation of the human user as teacher or mediator (Seabra Lopes & Wang, 2002; Steels & Kaplan, 2002).

A learning system in a robot should support long-term learning and adaptation, as is common in animals and, particularly, in humans. For that purpose, the learning system should exhibit several basic properties (Seabra Lopes & Wang, 2002), namely:

- *Supervised* — to include the human instructor in the learning process. This is an essential property for supporting the external/social component of symbol grounding.
- *On-line* — so that learning takes place while the agent is running.
- *Opportunistic* — the system must be prepared to accept a new example when it is observed or becomes available, rather than at pre-defined times or according to a pre-defined training schedule. This is another essential property for complying with the dynamics underlying external grounding.
- *Incremental* — it is able to adjust the learned descriptions when a new example is observed.
- *Concurrent* — it is able to handle multiple learning problems at the same time.
- *Meta-learning* — it is able to determine which learning parameters are more promising for different problems, ensuring each problem is handled efficiently.

With respect to the specific learning technique or paradigm, symbol grounding involves finding the invariant perceptual properties of the objects or categories to which symbols refer (Barsalou, 1999; Harnad, 1990). This suggests that learning of symbol meanings should be (predominantly) based on positive examples. Additionally, it should be noted that it is not easy to provide counter-examples in open-ended domains like word learning. Learning from positive examples is the basis for the one-class learning paradigm (Japkowicz, 1999; Tax, 2001), which was adopted for the work described below, and in some previous work (Wang & Seabra Lopes, 2004).

Architecture

The whole system comprises two main components (Figure 1), namely the artificial agent (for historical reasons here called the Student) and its World (including the human Instructor). The agent architecture itself consists of a perception system, an internal lifelong learning and classification system (OCLL) and a limited action system. At present, the action system abilities are limited to reporting the classification results back to the Instructor. Since the current agent perceives and acts on the physical world, it will also be referred to as a robot.

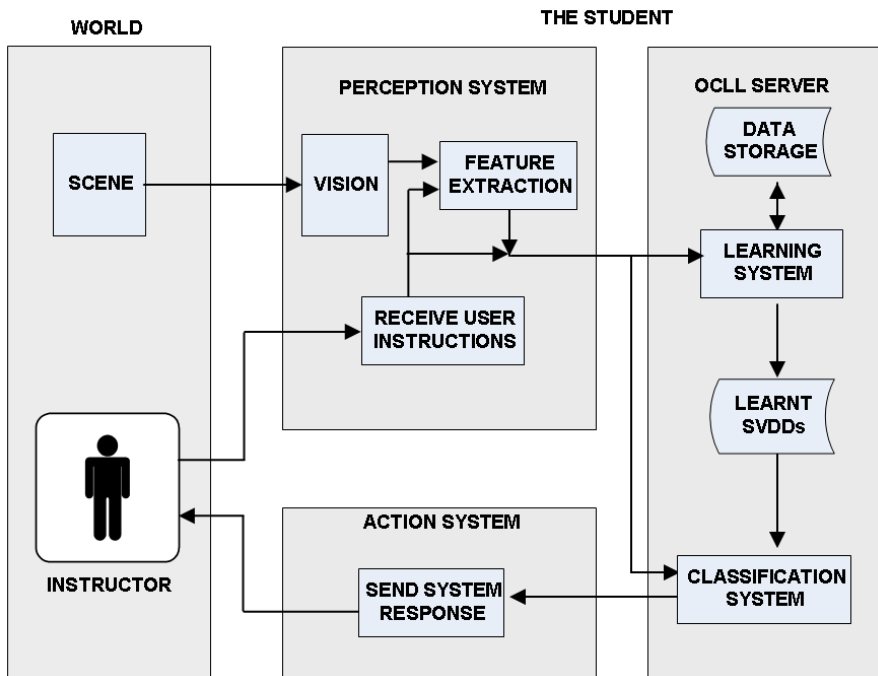


Figure 1. System Architecture

Instructor and the world

The world includes the user, a visually observable area and real-world objects (e.g., pen, stapler, mobile, mouse) whose names the instructor may wish to teach. The user, who is typically not visible to the robot, will act as instructor or mediator. As instructor, he or she has the role of communicating with the robot. Using a simple interface, the instructor can select (by mouse-clicking) any object from the robot's visible scene, thereby enabling shared attention. Then, the instructor can perform the following actions:

- Teach the object's class name for learning
- Ask the class name of the object, which the robot will determine based on previously learned knowledge
- If the class returned in the previous case is wrong, the instructor can send a correction.

The student

The student robot currently is a computer with an attached camera.¹ The computer runs the visual perception and learning/classification software as well as the communication interface for the instructor. The tasks of the perception system include capturing images from the camera, receiving instructions from the user (sending objects for either learning or classification) and extracting object features from images (Figure 1).

Once the user points the mouse to the desired object in the image, an edge-based counterpart of the whole image is generated.² From this edges image, the boundary of the object is extracted taking into account the user-pointed position.³

The boundary image contains all pixels located at the boundary edges of the object. Figure 2 illustrates the stated stages of pre-processing to extract the boundary image of the object class *Stapler*. At this point, the instructor can check whether the extracted boundary image adequately represents the object and decide whether to use it for learning or classification.

Objects should be described to the learning algorithm in terms of a small set of informative features. A small number of features will shorten the running time for the learning algorithm. Information content of the features will determine the learning performance. For visual object recognition in an artificial/robotic system, it seems crucial that features capture the object's shape. As mentioned earlier, children show a strong attentional bias towards shape when learning names of artifacts. Moreover, shape should be captured independently of position and orientation in the scene.



Figure 2. Image pre-processing stages in extracting the boundary of the object *Stapler* from the original image.

(Left: the original scene; center: the edges image; right: the boundary image of *Stapler*)

To meet these requirements, a feature extraction strategy was devised that captures the variation of the distance of boundary pixels to the center of the object. For this purpose, the smallest circle enclosing the object is divided into 36 sections of 10° . Each section i contains a number of boundary pixels with angle θ_i , such that $10 \times (i - 1) \leq \theta_i < 10 \times i$. The average distance of these pixels to the center of the circle, a radius R_i , is computed. Based on the R_i values, the following features are then computed:

- Radius average, R — the average of all R_i .
- Radius standard deviation, S — again computed over all R_i .
- Normalized radii, r_i — this is a vector containing the normalization of all R_i values with respect to the average radius R , but rotated to make it orientation-invariant. It is computed in two steps:
 - First, the normalized values are computed as $r_i = R_i/R$.
 - Then, all values are rotated in the vector in such a way that highest values are at the center, according to a local average measure. Specifically, a given section i will be at the center if the average of all values r_j , with $j = i - 4, \dots, i + 4$, is the highest.
- Normalized radius standard deviation, s — computed over all r_i .
- Block averages, B_k — the normalized radius values are divided into six blocks; for each block k , where $k = 1, \dots, 6$, B_k is defined as the average of all r_i values, for $i = (k - 1) \times 6 + 1, \dots, k \times 6$.

This feature extraction strategy provides 45 features to the learning algorithm. The first 2 features (R and S) provide size information. The remaining 43 normalized features capture the shape of a segmented object, invariant to its size, translation and rotation. Figure 2 (left) shows a scene with three objects (a stapler, a pen and a ball). Figure 3 shows the normalized radius vectors for the three objects.

This method is an original proposal of the authors; one of its main advantages is that it is simple to implement. The histogram approach of Roy and Pentland (2002), of which we were initially unaware, although quite different, also seems straightforward to implement. We intend to compare the two approaches in the near future.

The communication between the student robot and human instructor is supported by the perception and action systems (for instructor input and robot feedback, respectively). 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 and classification capabilities are provided to the agent using a client-server approach. A new learning server, implemented as a separate process,

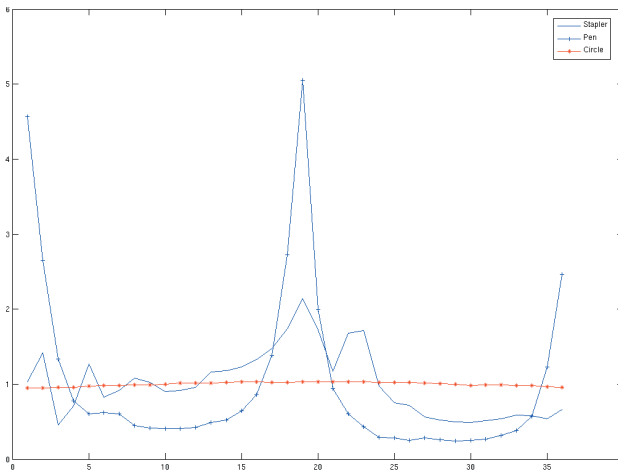


Figure 3. Normalized and rotated radius feature vectors for the three objects in Figure 2

performs both lifelong learning and classification as requested by the user. Its importance in this work is shown in the next section, which presents its design and functionality.

One-class lifelong learning

As mentioned above, one-class learning is an interesting candidate learning paradigm for such an open-ended domain as word learning. Tax (2001) describes and experimentally compares a large number of methods from the perspective of one-class learning, including Parzen density estimator, nearest neighbor, autoassociators, SVDD, LVQ, PCA, SOM, k-means and k-centers. One of the preferred methods is Support Vector Data Description (SVDD), a method that shares its foundations with the support vector classifier (Vapnik, 1995). It shows one of the best performances, provided that the training instances are not too few (e.g., less than 10). Usually, the performance on the training set is a useful indication of the performance on the test set. The performance is especially good when the training distribution is different from the target distribution. Furthermore, SVDD is particularly good at avoiding overfitting. Finally, the evaluation time is very small. We therefore selected SVDD as the base for the developed OCLL⁴ system.

OCLL decomposes into two concurrent threads of processing (Figure 4). The *main thread*, supports communication with the learning client (the agent) and also runs the classification routines. The other, the *learning thread*, determines learning parameters and runs the SVDD algorithm. Having separate threads for learning and classification allows the OCLL server to execute client requests for

classification and save new data for learning, while the learning thread concurrently handles learning.⁵

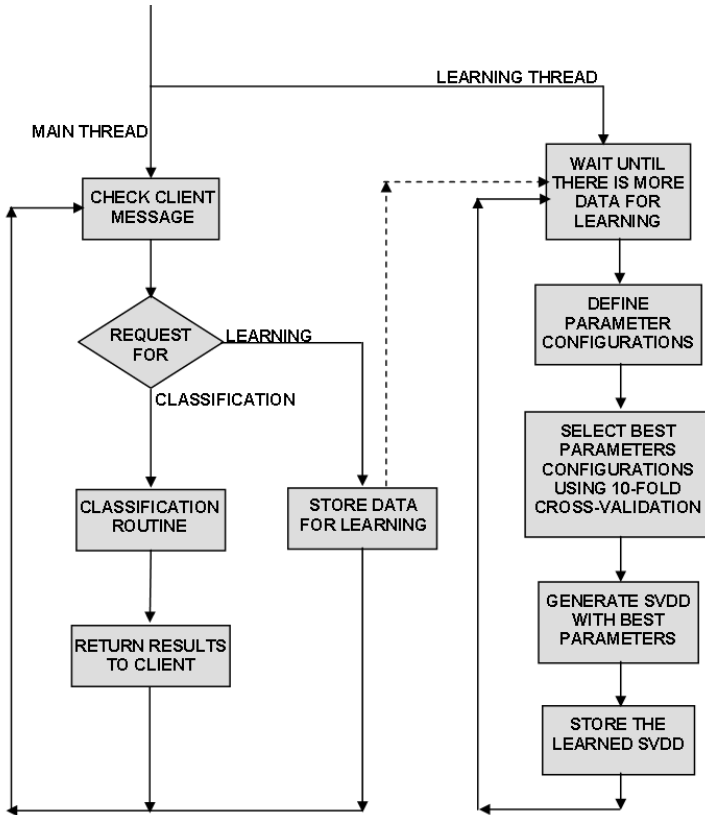


Figure 4. Flow chart of OCLL (dashed line indicates sequencing across the two different threads)

Learning

In the normal case, SVDD is trained only with positive instances of the target class. It tries to form a hypersphere around the data by finding an optimized set of support vectors. These support vectors are data points on the boundary of a hypersphere whose center is also determined through optimization. The hypersphere’s center is assumed to represent the center of the data distribution itself. If outliers (negative instances) are available, they add to the performance of SVDD since, during optimization, an even tighter boundary around the data can be obtained. An introduction to SVDD’s underlying mathematics is given in the Appendix.

The configuration parameters to be supplied to the SVDD algorithm are the percentage of training objects that can be considered as outliers (for better boundary description and over-fitting avoidance), *FRACREJ*, and the width parameter of the Gaussian kernel, s (used to map the data into a more suitable space). OCLL tries several values of *FRACREJ* and s and uses those with the highest performance for learning the final class description.

In the current implementation of OCLL, values of *FRACREJ* range from 1% to 11% of the training data, with an interval of 2%. This means that a total of 6 values will be tested. The maximum and minimum distance between any two training objects determines the range of values of the width parameter (for a thorough explanation on the range of values and the choice of s refer to Tax, 2001):

$$\min \|x_i - x_j\| \leq s \leq \max \|x_i - x_j\|$$

Since the magnitude of object features can vary significantly, the above range of values of s is divided into 10 parts on a logarithmic scale. This leads to 11 possible s values. In total, there are 66 combinations of values of *FRACREJ* and s for evaluation.

OCLL performs cross-validation to determine the best configuration. Specifically, for each parameter setting, 90% of the examples (randomly selected) are used for training and the remaining examples are used for performance evaluation. This is repeated 10 times, and the average performance is retained.⁶ The following performance measure, which combines precision and recall values, is used:

$$\frac{2 \cdot P \cdot R}{P + R}$$

where, $P = CTP/TP$ is precision, $R = CTP/TE$ is recall, CTP is the number of correct target predictions, TP is the number of target predictions and TE is the number of target examples.

As mentioned previously, learning is incremental and supervised. Thus, when an object is misclassified, the instructor has the option of providing the correct class, so that class descriptions can be improved. Misclassification in this case can be of two types: either the object is inside the hyperspheres of several classes and OCLL chose the wrong class, or the object is outside the hyperspheres of all known classes. Given a correction from the user, OCLL will identify and retrain the class descriptions needing correction. Specifically, OCLL will add the misclassified object as outlier for retraining the classes whose hyperspheres contain the object.

Classification

In the standard application of SVDD class descriptions, a new object is classified as belonging to the target class if it is determined to be inside the hypersphere of the class. Using this criterion in OCLL, more than one class or none of the classes might be identified as the target, and a classification decision would be impossible to make. For this reason, a more suitable criterion has been adopted. In particular, a distance metric, called “Normalized Distance to the Center” (*NDC*) was introduced. For a given object z , $NDC(z)$ is the distance of z to the center of the hypersphere given as a fraction of its radius. It captures the relative closeness of the object from the center of each class and, therefore, enables comparison of its membership to different classes. For a particular class, the lower the value of $NDC(z)$, the closer the object is to the centre of that class. Of all the classes that have been learned, the one with the lowest $NDC(z)$ will be considered as best class candidate for object z . However, if the lowest value of $NDC(z)$ is greater than 2.0, the object is considered to be clearly outside any of the current class descriptions and thus not belonging to any class.

Experimental methodology

The word learning research surveyed above has some common features. One of them is the limitation on the number of learned words: the described 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, such as semantic accuracy (Roy & Pentland, 2002), classification success (Steels & Kaplan, 2002), word-meaning grounding accuracy and object categorization accuracy (Yu, 2005). Some authors plot this type of measures versus training time. 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.

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. The following aspects of a long-term category learning process should therefore be considered:

- Evolution: Depends on the ability of the learner to adjust category representations when a new word is introduced.
- Recovery: The discrimination performance will generally deteriorate with the introduction of a new word. The time spent in system evolution until correcting and adjusting all current categories defines recovery. Recovery is based on classification errors and corresponding corrective feedback.
- Breakpoint: Inability of the learner to recover and evolve when a new category is introduced.

A well-defined teaching protocol can facilitate the comparison of different approaches as well as the assessment of future improvements. With that in mind, along with the evolution, recovery and breakpoint aspects just described, the teaching protocol of Figure 5 is proposed.

```

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 5. Teaching protocol used for performance evaluation

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 $3 \times n$, 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. However, the evaluation/correction phase continues until a local maximum is reached.

Experimental results

Experiments were conducted according to the protocol proposed above. The set of words (object category names) and the set of training objects were not established in advance. As categories were learned, new objects were fetched from the surrounding office environment and used to introduce new categories. In the first experiment (already presented in Seabra Lopes & Chauhan, 2006), new categories were introduced in the following sequence:

Pen – 5 different pens were used for teaching
Stapler – 1 object of this class
Ball – 2 circular objects
Mobile – 3 objects
Key – 2 objects
Box – 2 objects
TiltedCup – 1 object
Rubber – 1 object
CoffeeCup – 1 object
StapleRemover – 1 object

In later experiments, these objects and categories were used again, in different sequences. In some experiments, it was possible to introduce two additional categories, bringing the total to 12:

ScrewDriver – 1 object
Plug – 1 object

Figure 6 shows instances of the used object classes. Natural light variation significantly affects the quality of the images collected from the camera, because the scene is just in front of a window.

Obtained results are graphically presented in Figures 7 and 8. They respectively show the evolution of *classification precision* and *learning efficiency* against the number of question/correction iterations. Efficiency is defined as the ratio between the obtained classification precision and the precision of a random classifier. In each iteration, the precision of the random classifier is computed based on the number of currently introduced classes. Points of high instability of the measures in Figures 7 and 8 in most cases indicate the introduction of a new class.

Classification for an object of the first class (*pen*) was correct in the first attempt. This means a single iteration was enough to reach a precision of 100%. Similarly, for the second class, in the minimum number of iterations (at least n iterations for n classes, as defined above) maximum precision was obtained. On the introduction of the third class (*ball*), although starting at 100%, precision

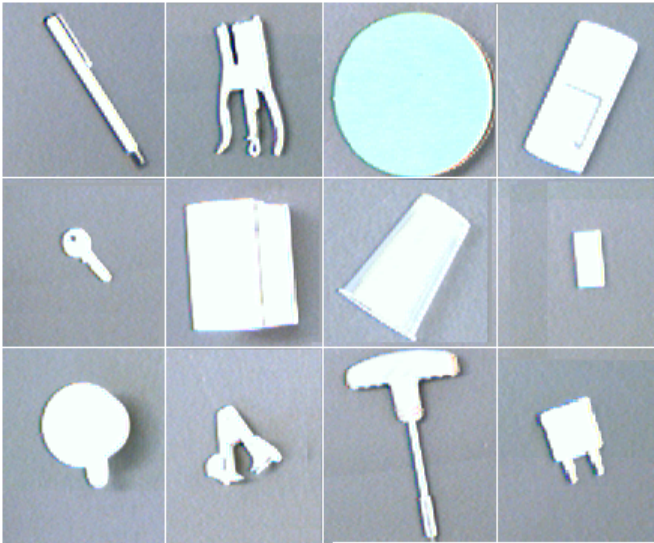


Figure 6. Instances of object classes used in experiments

continuously dropped to 50%, before it recovered to a value above the threshold.⁷ For the next classes, the pattern remained similar: precision degrades at the introduction of the new class and then recovers after a number of question/correction iterations.

On the introduction of the 10th class (*staple remover*), precision started at 100%, then dropped to values between 20% and 50%, remaining there for many iterations. As can be seen at the end of the graphs of Figures 7 and 8, classification precision and learning efficiency seem to have stabilized. No considerable improvement in these measures could be noticed over time. Here the instructor concluded that, on the extracted set of features and for the above set of classes, the learning capacity of the student had reached its breakpoint.

It should be noted that, most of the time, learning efficiency is above 2.0, and its average is 4.3. This means that precision is significantly above the random classifier precision throughout the whole experiment. Another important observation can be made. While classification precision seems to follow a decreasing trend as the number of introduced classes increases, learning efficiency follows an increasing trend almost until the breakpoint class.

In OCLL, each correction leads to the introduction of new positive and negative examples. For each new class introduced, Figure 9 shows the total number of outliers and positive examples added to the system to achieve the precision threshold. It can be seen from this figure that introduction of the last two classes adds high numbers of outliers as well as positive examples. In comparison to the initial eight

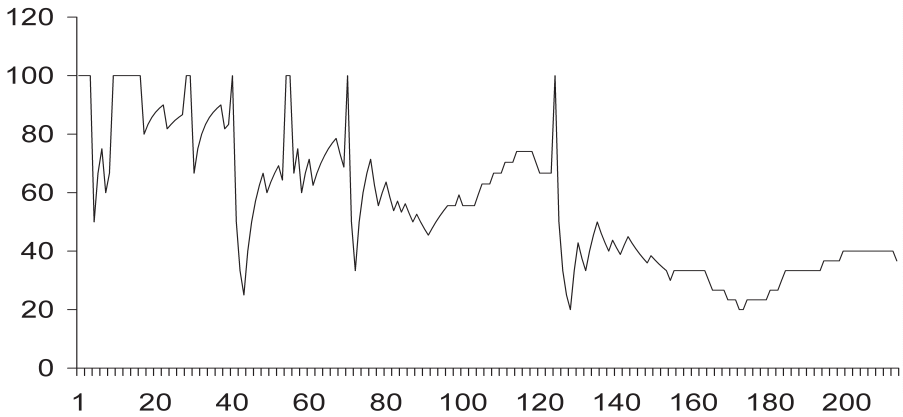


Figure 7. Evolution of classification precision versus number of question/correction iterations

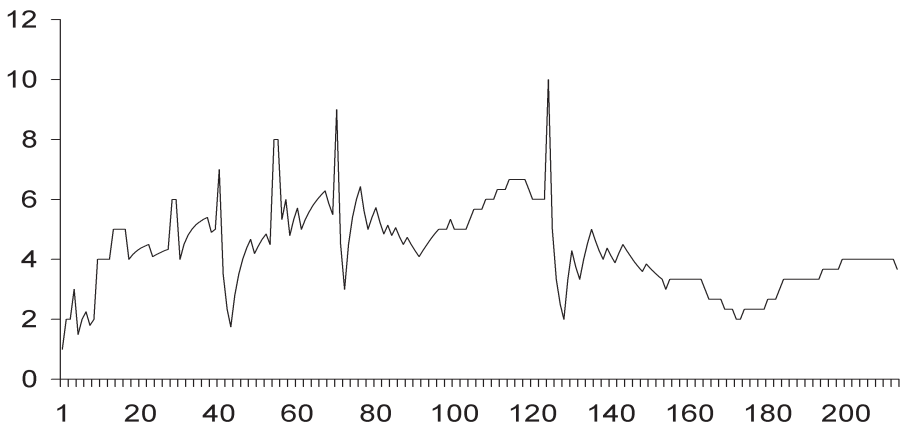


Figure 8. Evolution of learning efficiency versus number of question/correction iterations

classes, the number of misclassifications after the 9th class was introduced shows a substantial increase. In other words, it became increasingly difficult for the system to reach the precision threshold. Eventually, on the 10th class the system reached its breakpoint. A collective analysis of Figures 7–9 shows an association between the number of iterations required for reaching the precision threshold and the number of outliers and positive examples needed before the system reached the precision threshold. For the first eight classes, the system shows fast evolution of precision and efficiency. The number of examples and outliers that was necessary to add after introducing those classes are also relatively few. On the other hand, for the 9th and 10th classes, it took a long time to reach the precision threshold (not

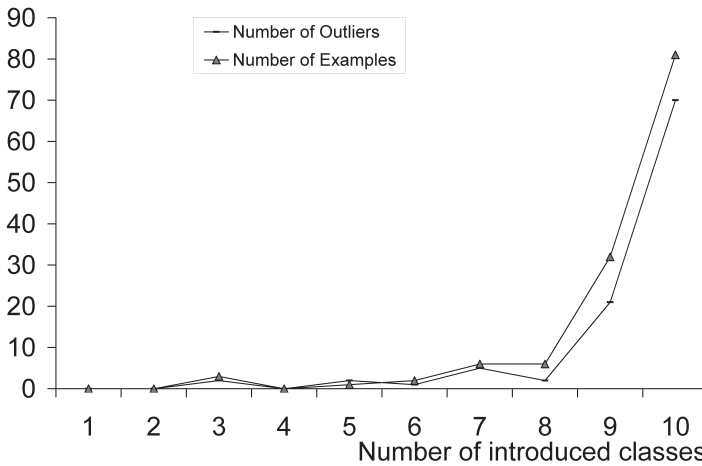


Figure 9. Number of outliers and positive examples added after each new class was introduced in the experiment of Figures 7–8

achieved after introducing the 10th class) and the number of outliers and target examples introduced to the system showed a sharp increase.

Table 1 shows the number of outliers and positive examples stored for each class after the system reached the breakpoint. As can be observed, the number of outliers in some classes (*Box* and *TiltedCup*) far outweighs the number of target examples.

Table 1. Final number of target and outlier examples in each class (for the experiment of Figures 7–8)

Object class	Target	Outliers
Pen	17	18
Stapler	24	1
Ball	30	7
Mobile	27	1
Key	22	1
Box	14	49
TiltedCup	17	40
Rubber	24	1
CoffeeCup	19	1
StapleRemover	14	9

Tax (2001) observed that introducing a few outliers to the training data results in better class descriptions for one-class classifiers, but introducing too many deteriorates learning performance. For the first introduced classes, classification precision improved very quickly, which supports the idea that, when using few outliers,

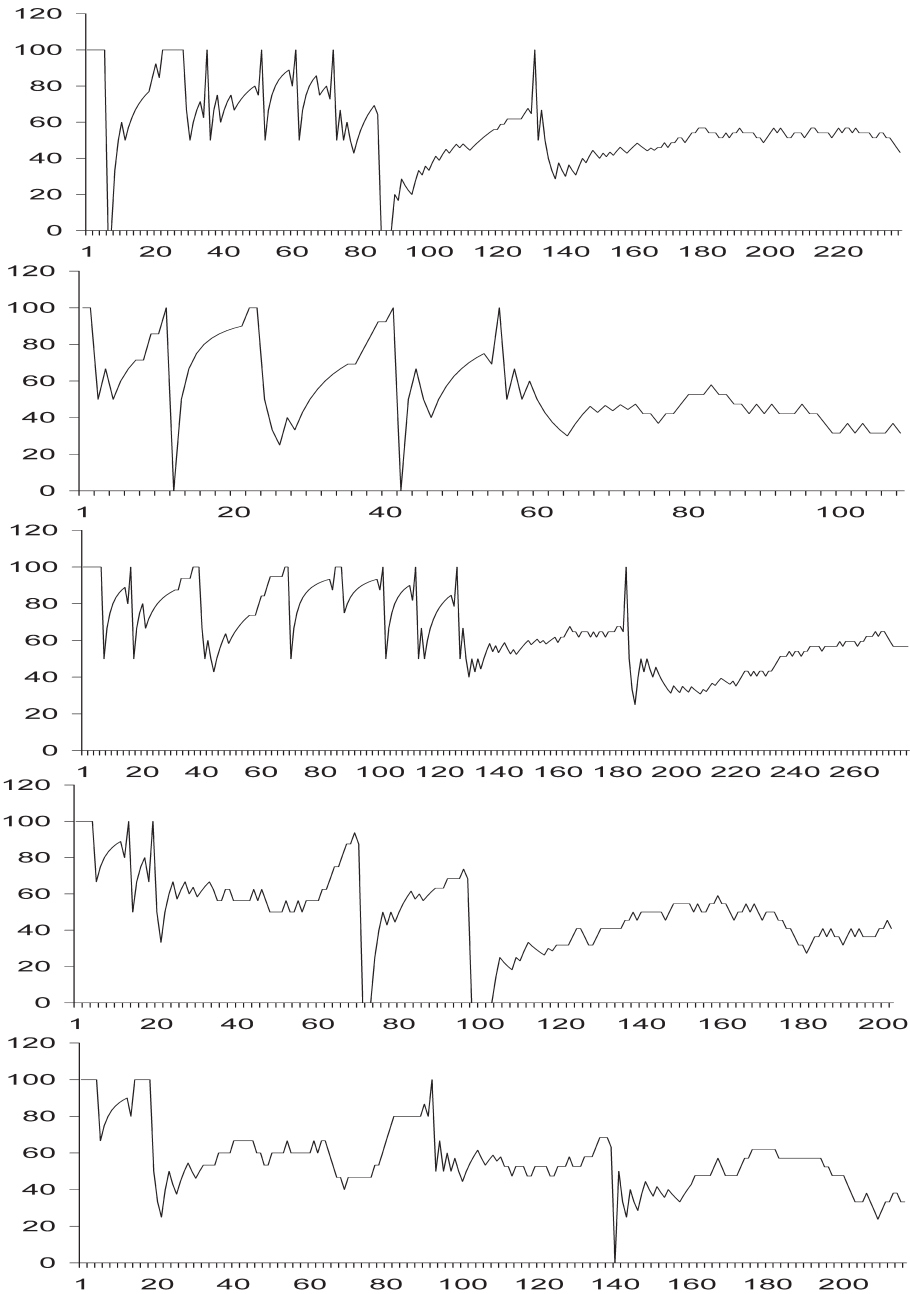


Figure 10. Evolution of average classification precision in the performed experiments. The horizontal axis is the number of question/correction iterations and vertical axis is precision. The number of introduced words is respectively: 10, 12, 6, 12, 7 and 7 (Figure 7 and then Figure 10 from top to bottom)

SVDD class descriptions do describe the data better than just with positive instances. However, in the long run, the number of outliers in the training data may become higher than the number of target examples. This may also limit the number of classes that the system will be able to learn.

The experiment was repeated several times (Figure 10). The object classes (and therefore the words) were the same, although some of the instances were different. The experimental set up also varied slightly, since the camera is now at a greater distance from the objects, which implies less resolution. In the first three of these experiments, the same sequence of introduction of new classes was used. Breakpoint was detected after 6 introduced classes in one case (109 iterations) and after 12 classes in the other two cases (239 and 279 iterations). Two additional experiments were carried out for different (randomly generated) sequences. In both, it was possible to reach only 7 classes (203 and 217 iterations). These experiments (Figure 10) are in line with the observations already discussed for the initial experiment.

Summary and discussion

This paper addresses word learning for human–robot interaction. Instead of following a computationalist approach strictly focused on finding invariances for category learning, a more integrational approach was adopted, which is in line with previous work in our group (Seabra Lopes, 2002; Seabra Lopes & Connell, 2001b; Seabra Lopes et al., 2005). In this approach, the robot's user is explicitly included in the language acquisition process. Through mechanisms of shared attention and corrective feedback, the human user, acting as instructor or mediator, can help the agent ground the words used to refer to objects that it finds in its environment. This is consistent with current views on distributed cognition, distributed language and external symbol grounding (Cowley, 2007a).

The present work was carried out with a particular concern for the fact that word learning is an open-ended domain. This means it cannot be realistic to address the problem as if there is a pre-defined set of words, although that is the typical approach to the problem. A vocabulary can grow as long as the perceptual and cognitive abilities of an agent allow it to grow. In characterizing vocabulary growth, aspects of evolution of category representations, recovery from confusion after the introduction of new words and breakpoint are important. Based on these considerations, a novel experimental methodology was proposed for evaluating a word learning method, from the first acquired words until the limits of the method. This methodology can be useful for comparing the word learning capabilities of different agents as well to assess research progress with respect to scaling-up to larger vocabularies.

While not strictly focused on the problem of finding invariances for category learning, this problem was addressed with a new approach based on the *one-class learning* paradigm and, in particular, on support-vector data descriptions. More importantly, the learning aspects were also addressed at a more integrational level. A learning module, OCLL, acting as “learning server” for category formation and evolution, was implemented and integrated in the agent. The way OCLL works has several important properties: It is supervised, on-line, opportunistic, incremental, concurrent and capable of meta-learning. Since all of these properties are required for the learning computations, some of them (e.g., supervised, opportunistic) are crucial to support the dynamics of external symbol grounding.

The proposed experimental methodology was used to evaluate the performance of the agent on learning the names of several real-world office objects. From the conducted experiments, it can be concluded that the agent has the ability to incrementally evolve to include each new word presented. Classification precision falls after the introduction of new words, but quickly recovers by correcting the classifiers so that the boundaries of the respective class descriptions get modified to separate out all the different classes.

As the number of words increases, the training becomes more difficult and some class descriptions have to be corrected many times before the precision threshold is achieved. Eventually, the learning capacity of the agent reaches its breakpoint. In the different runs of experiments, this happened between the 6th and the 12th word. Although the different published works on word learning are not directly comparable to each other or to the above presented work (due to their many specificities), the results reported above are comparable to results previously reported by others with respect to the number of learned words.

These results raise several issues for discussion. The main issues are concerned with the existence of a breakpoint in the word learning capacity of a robot. The existence itself seems easy to accept. Robots are limited in their perceptual (sensors, sensor fusion, active sensing) and sensorimotor abilities and, therefore, cannot learn arbitrarily large numbers of categories, particularly when perception and action do not enable them to detect small between-category differences.

The fact that the breakpoint occurs no later than the 12th word is more problematic. A robot with such a limitation will not be of any use in environments that require language-based interaction with users. Why can't these systems learn more words and categories? In particular, why can't they learn more concrete object categories and names?

A combination of the following factors may explain the limitation:

- sensor limitations
- lack of active sensing/animate vision

- lack of physical interaction with the target objects
- lack of consideration of the affordances of objects
- limited interaction between the learning agent and the human caregiver
- inappropriate perceptual abilities and representations
- inappropriate category learning methods.

With respect to sensing, our agent is very limited. A single camera enables perception of target objects, and it remains in a fixed position over the objects. There is no possibility either for active/animate sensing or for sensorimotor experience with the objects. This limitation is common to most existing prototypes and models. The most notable exception is reported by Roy and Pentland (2002), who use a robotic arm and a turn-table to capture multiple views of each object. None of the surveyed systems uses physical interaction with objects to categorize them.

Moreover, because of these limitations, neither our system nor the surveyed systems are able to take into account the affordances of objects, that is, the actions and uses that they afford. Interestingly, an early explanation for the shape bias, observed in children when learning concrete object names, was based on the conjecture that shape would be a strong determiner of affordances (Rosch, 1973). This is currently a subject of investigation (Gershoff-Stowe & Smith, 2004). Gibson (1979) also stressed the importance of affordances in visual perception. An early robotic model that categorizes affordances (the produced categories are here called proto-symbols) and uses them to guide navigation was described by MacDorman, Tatani, Miyazaki and Koeda (2000).

Our approach explicitly includes the human user as instructor or mediator for the word learning process. However, in the current version, the initiative for interaction between the instructor and the artificial agent is always on the instructor's side. In the foreseeable future, we intend to integrate a robotic arm into the agent and assign it goals. In this richer scenario, the agent may also wish to ask for instructor's help to the language acquisition process. This results in a mixed-initiative interaction that allows for dual control and mutual gearing, as observed in the relation between infant and caregiver (Cowley, 2007a, 2007b).

The limitations discussed so far are concerned with the "external" component of language acquisition and symbol grounding. Meanwhile, the internal mechanisms should also be significantly improved. The learning approaches that have been used until now in most systems are either connectionist or instance-based. In the work reported above, one-class learning with support-vector data descriptions was used. All these approaches take as input collections of feature vectors representing training instances. However, it has been pointed out that feature vectors are not always the best representation for learning (Aha & Wettschereck, 1997). This is particularly the case when instances can be split into components leading to a structured/relational

representation. Object categorization based on shape, as we have implemented (and is the key bias in children as well), could be better handled with relational instead of vector-based representations. That is also the assumption underlying the cognitive theory of recognition-by-components (Biederman, 1987). These ideas link to some of the literature on symbol systems and symbol grounding. For instance, the theory of “perceptual symbol systems” (Barsalou, 1999) emphasizes that a perceptual symbol represents a schematic component of a perception, and not a holistic experience. Also, the “Symbolic Theft Hypothesis” (Cangelosi & Harnad, 2000) emphasizes that complex categories can be learned more efficiently from more basic categories than directly from sensor data. Based on all these considerations, we believe that adopting component-based/relational representations is a promising research path. In this approach, more complex, composed categories would represent individual physical objects instead of primitive categories, as is usually the case.

As can be seen from this discussion, there is plenty of work for the robotics, AI and cognitive science communities concerning the development of artificial agents able to acquire extended vocabularies. Our own work will focus on improving the internal mechanisms, as well as extending the sensorimotor and interaction capabilities of the robot.

Notes

* 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. The authors would like to thank Stephen Cowley, Karl MacDorman and Armando Pinho for enlightening discussions and anonymous reviewers for many constructive and helpful comments.

1. An IEEE1394 compliant *Unibrain Fire-i digital camera* is being used.
2. The implementation of the *canny algorithm*, from the publicly available openCV library of vision routines, is used for edge detection. Other openCV functions have been used in the implementation. See <http://www.intel.com/technology/computing/opencv/index.htm>.
3. This is performed using a region growing algorithm.
4. The name derives from “one-class lifelong learning.”
5. The OCLL server is a C++ program which runs as a single Linux process divided into two threads. An implementation of SVDD for MATLAB is in the dd-tools toolbox (Tax, 2005) publicly available at http://www-ict.ewi.tudelft.nl/~davidt/dd_tools.html. Therefore, in practice, the SVDD algorithm runs in a separate MATLAB-based process on request of the learning thread of OCLL. The main thread saves any new training data in a file and informs the learning thread to process it. When there is no new data to process, the learning thread is waiting on a semaphore.

When new data is received, the learning thread calls SVDD on the MATLAB process and waits until SVDD returns. The MATLAB process stores the learned class description in a file.

6. For this reason, the instructor must provide at least 10 examples for learning to start.
7. In these experiments, the precision threshold was set to 0.667. This ensures that, in a stable situation, there will be at least twice as many correct answers as incorrect answers, which intuitively appears to be a suitable baseline for acceptable performance.

References

- Aha, D. W. & Wettschereck, D. (1997). Case-based learning: Beyond classification of feature vectors. In M. van Someren & G. Widmer (Eds.), *Proceedings of the Ninth European Conference on Machine Learning (ECML 1997)* (pp. 329–336), LNCS 1224, London, UK: Springer-Verlag.
- Ameel, E., Malt, B., & Storms, G. (2006). Object naming and later lexical development. In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (pp. 18–23). Mahwah, NJ: Lawrence Erlbaum.
- Anderson, M. L. & Perlis, D. R. (2002). Symbol systems. In Nadel, L., Chalmers, D., Culicover, P., French, B. & Goldstone, R., *Encyclopedia of Cognitive Science*, London, UK: Macmillan.
- Barsalou, L. (1999). Perceptual symbol systems, *Behavioral and Brain Sciences*, 22(4), 577–609.
- Bates, E., Thal, D., Finlay, B. & Clancy, B. (2002). Early language development and its neural correlates. In S. J. Segalowitz & I. Rapin (Eds.), *Handbook of neuropsychology: Child neuropsychology* (Vol. 7, pp. 69–110). Amsterdam: Elsevier.
- Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94, 115–147.
- Breazeal, C. & Scassellati, B. (2000). Infant-like social interactions between a robot and a human caregiver. *Adaptive Behavior*, 8(1), 49–74.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence Journal*, 47, 139–159.
- Cangelosi, A. (2005). Approaches to grounding symbols in perceptual and sensorimotor categories. In H. Cohen & C. Lefebvre (Eds.), *Handbook of Categorization in Cognitive Science* (pp. 719–737). Oxford: Elsevier Science.
- 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.
- Clark, A. (1997). *Being there: Putting brain, body and world together again*. Cambridge, MA: MIT Press.
- Chatila, R. (2004). The cognitive robot companion and the European ‘Beyond robotics initiative’. In *Proceedings of the Sixth EAJ International Symposium on Living with Robots*. Tokyo, Japan.
- Cowley, S. J. (2007a). Distributed language: Biomechanics, functions and the origins of talk. In C. Lyon, C. Nehaniv & A. Cangelosi (Eds.), *Emergence of communication and language*. Heidelberg (pp. 105–127), London: Springer.
- Cowley, S. J. (2007b). How human infants deal with symbol grounding. *Interaction Studies*, 8(1), PPP-PPP.

- Crystal, D. (1987). How many words? *English Today*, 12, 11–14.
- Fong, T., Nourbakhsh, I. & Dautenhahn, K. (2003). A survey of socially interactive robots: Concepts, design, and applications, *Robotics and Autonomous Systems*, 42, 143–166.
- Gershoff-Stowe, L. & Smith, L. B. (2004). Shape and the first hundred nouns. *Child Development*, 74(4), 1098–1114.
- Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin.
- Gillette, J., Gleitman, H., Gleitman, L. & Lederer, A. (1999). Human simulations of vocabulary learning. *Cognition*, 73, 135–176.
- Greco, A., Riga, T. & Cangelosi, A. (2003). The acquisition of new categories through grounded symbols: An extended connectionist model. O. Kaynak, E. Alpaydin, E. Oja and L. Xu (Eds.), *Artificial Neural Networks and Neural Information Processing — ICANN/ICONIP 2003*, Springer, pp. 773–770.
- Harnad, S. (1990). The symbol grounding problem, *Physica D*, 42, 335–346.
- Harnad, S., Hanson, S. J. & Lubin, J. (1991). Categorical perception and the evolution of supervised learning in neural nets. In D. W. Powers & L. Reeker (Eds.), *Working Papers of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, pp. 65–74.
- Harnad, S., Hanson, S. J. & Lubin, J. (1995). Learned categorical perception in neural nets: Implications for symbol grounding. In V. Honavar & L. Uhr (Eds), *Symbol processors and connectionist models in artificial intelligence and cognitive modelling: Steps toward principled integration* (pp. 191–206). San Diego, CA: Academic Press.
- Japkowicz, N. (1999). Are we better off without counter-examples? In *Proceedings of the First International ICSC Congress on Computational Intelligence Methods and Applications (CIMA-99)*, pp. 242–248.
- Levinson, S. E., Squire, K., Lin, R. S. & McClain, M. (2005). Automatic language acquisition by an autonomous robot. *Proceedings of the AAAI Spring Symposium on Developmental Robotics*. March 21–23.
- Love, N. (2004). Cognition and the language myth. *Language Sciences*, 26, 525–544.
- MacDorman, K. F., Tatani, K., Miyazaki, Y. & Koeda, M. (2000). Proto-symbol emergence, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, Takamatsu, Japan, pp. 1619–1625.
- Plunkett, K. & Sinha C. G. (1992). Connectionism and developmental theory. *British Journal of Developmental Psychology*, 10, 209–254.
- Plunkett, K., Sinha, C., Moller, M. F. & Strandsby, O. (1992). Symbol grounding or the emergence of symbols? Vocabulary growth in children and a connectionist net. *Connection Science*, 4(3–4), 293–312.
- 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. & Pentland, A. (2002). Learning words from sights and sounds: A computational model. *Cognitive Science*, 26, 113–146.
- Rosch, E. (1973). On the internal structure of perceptual and semantic categories. In T. E. Moore (Ed.), *Cognitive development and the acquisition of language* (pp. 111–144). San Diego, CA: Academic Press.
- Russell, S. & Norvig, P. (2003). *Artificial intelligence: A modern approach* (Rev. ed.). Upper Saddle River, NJ: Prentice-Hall.

- Sales, N. J. & Evans, R. G. (1995). An approach to solving the symbol grounding problem: Neural networks for object naming and retrieval. *Proceedings of the International Conference on Cooperative Multimodal Communications (CMC-95)*. Eindhoven, The Netherlands.
- Samuelson, L. & Smith, L. B. (2005). They call it like they see it: Spontaneous naming and attention to shape. *Developmental Science*, 8(2), 182–198.
- Seabra Lopes, L. (2002). Carl: From situated activity to language-level interaction and learning. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)* (pp. 890–896). Lausanne, Switzerland.
- Seabra Lopes, L. & Chauhan, A. (2006). One-class lifelong learning approach to grounding. *Workshop on External Symbol Grounding. Book of Abstracts and Papers* (pp. 15–23). University of Plymouth, Plymouth, UK.
- Seabra Lopes, L. & Connell, J. H. (Eds.) (2001a). *Semisient Robots* special issue of *IEEE Intelligent Systems*, 16(5).
- Seabra Lopes, L. & Connell, J. H. (2001b). Semisient robots: Routes to integrated intelligence. *IEEE Intelligent Systems* (special issue on *Semisient Robots*, L. Seabra Lopes & J.H. Connell, Eds.), 16(5), 10–14.
- Seabra Lopes, L. & Wang, Q. H. (2002). Towards grounded human–robot communication. In *Proceedings of 11th IEEE International Workshop on Robot and Human Interactive Communication (Ro-Man 2002)*, pp. 312–318.
- Seabra Lopes, L., Teixeira, A. J. S., Quinderé, M. & Rodrigues, M. (2005). From robust spoken language understanding to knowledge acquisition and management. *Proceeding of Inter-speech 2005* (pp. 3469–3472). Lisbon, Portugal.
- Smith, L. B. & Samuelson, L. (2006). An attentional learning account of the shape bias. *Developmental Psychology* 42(6), 1339–1343.
- Steels, L. (2001). Language games for autonomous robots. *IEEE Intelligent Systems* (special issue on *Semisient Robots*, L. Seabra Lopes and J. H. Connell, Eds.), 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.
- Talmy, L. (2000). *Toward a cognitive semantics: Concept structuring systems (language, speech and communication)*. Cambridge, MA: The MIT Press.
- Tax, D. M. J. (2001). *One-class classification: Concept learning in the absence of counter-examples*. Unpublished doctoral dissertation, Technische Universiteit Delft, The Netherlands.
- Tax, D. M. J. (2005). *DD Tools — The Data Description Toolbox for MATLAB. Version 1.4.1*, Technische Universiteit Delft, The Netherlands.
- Thrun, S. (1996). *Explanation-based neural network learning: A lifelong learning approach*. Boston, MA: Kluwer.
- Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59, 433–460.
- Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.
- Wang, Q. & Seabra Lopes, L. (2004). Visual object recognition through one-class learning. *Proceedings of International Conference on Image Analysis and Recognition (ICIAR 2004)*, Part I, LNCS 3211, Springer, pp. 463–469.
- Yoshida, H. & Smith, L. B. (2005). Linguist cues enhance the learning of perceptual cues. *Psychological Science*, 16 (2), 90–95.
- Yu, C. (2005). The emergence of links between lexical acquisition and object categorization: A computational study. *Connection Science*, 17(3–4), 381–397.

Authors' address

L. Seabra Lopes and A. Chauhan
Transverse Activity on Intelligent Robotics
IEETA/DETI, Universidade de Aveiro
Aveiro 3810-193, Portugal
lsl@det.ua.pt and aneesh.chauhan@ieeta.pt

About the authors

Luís Seabra Lopes is Assistant Professor at Universidade de Aveiro, Portugal, and leads the Transverse Activity on Intelligent Robotics at the university's Instituto de Engenharia Electrónica e Telemática. His interests include spoken language human–robot interaction, language acquisition, robot learning, multi-robot systems, and service robotics applications. He received a licenciatura degree in computer science in 1990, and a PhD in electrical engineering in 1998, both from Universidade Nova de Lisboa, Lisbon, Portugal. He is a member of the IEEE, a founding member of the Portuguese IEEE-RAS Chapter, and a member of the Portuguese AI Society (APPIA).

Aneesh Chauhan is Research Assistant at the Instituto de Engenharia Electrónica e Telemática of the Universidade de Aveiro. His research interests include statistical learning, pattern recognition, machine vision and language grounding. He received the B.Eng. in computer science and engineering from Babasaheb Ambedkar Marathwada University, Aurangabad, MH, India in 2001, and an M.Sc. in autonomous systems from the University of Exeter, UK, in 2004.

Appendix — Mathematical background

SVDD forms a hypersphere around the data by finding an optimized set of support vectors. These support vectors are data points on the boundary of a hypersphere whose center is also determined through optimization. The optimization process, that determines the center and support vectors, attempts to minimize two errors:

- Empirical error — percentage of misclassified training samples.
- Structural error — defined as R^2 , where R is the radius of the hypersphere, must be minimized with respect to center a and constraints $\|x_i - a\|^2 \leq R^2$, for every training object x_i .

In the ideal case (no noise), all training objects can be included in the hypersphere and therefore the empirical error will be 0. In practical applications, however, this may result in over-fitting. Better results can be obtained with not much extra computational expense if a kernel is introduced to get a better data description (Tax, 2001). In addition, if a set of outliers (negative instances) is known, it adds to the performance of SVDD since, during optimization, an even tighter boundary around the data can be obtained. From (Tax, 2001), the final error L to be optimized is given as:

$$L = \sum_i \alpha_i K(x_i, x_j) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$$

with the following constraints on Lagrange multipliers:

$$\forall i, 0 \leq \alpha_i \leq C$$

$$\sum_i \alpha_i = 1 \text{ and } a = \sum_i \alpha_i x_i$$

C gives the tradeoff between the volume of the description and the errors. The kernel K maps the data into a more suitable space. Although the choice of kernel is data dependent, in most applications a Gaussian kernel will produce good results. Tax (2001) gives a thorough explanation of the performance benefits of this kernel. It is defined as:

$$K(x_i, x_j) = \exp\left(\frac{-|x_i - x_j|^2}{s^2}\right)$$

where s is the width parameter of the kernel.

Class descriptions provide the support vectors and their respective α_i and s . In the standard application of SVDD class descriptions, the criterion for classifying any new object z as target is:

$$\sum_i \alpha_i K(z, x_i) > \frac{1}{2}(B - R)^2$$

where, $B = 1 + \sum_i \alpha_i K(x_i, x_i)$ and R is the radius.

In OCLL, given the need to decide which class is more suitable for a particular instance, it was necessary to define the following normalized distance metric:

$$NDC(z) = \frac{\sqrt{B - 2 \sum_i \alpha_i K(z, x_i)}}{R}$$

NDC (Normalized Distance to the Center) is the distance of an object z to the center of the hypersphere given as a fraction of its radius R .

