Connection Science
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713411269

Open-ended category learning for language acquisition

Luís Seabra Lopes [ab]; Aneesh Chauhan [a]
[a] IEETA, Universidade de Aveiro, Aveiro, Portugal [b] Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Aveiro, Portugal

Online Publication Date: 01 December 2008

To cite this Article Lopes, Luís Seabra and Chauhan, Aneesh(2008)'Open-ended category learning for language acquisition',Connection Science,20:4,277 — 297

To link to this Article: DOI: 10.1080/09540090802413228

URL: http://dx.doi.org/10.1080/09540090802413228

Connection Science
Vol. 20, No. 4, December 2008, 277–297

Open-ended category learning for language acquisition

Luís Seabra Lopes[a,b]* and Aneesh Chauhan[a]

[a]*IEETA, Universidade de Aveiro, Aveiro, Portugal;* [b]*Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Aveiro, Portugal*

(*Received 3 March 2008*)

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. To be more adaptive and to improve learning performance as well as memory usage, this learning architecture includes a metacognitive processing component. Multiple object representations and multiple classifiers and classifier combinations are used. At the object level, the main similarity measure is based on a multi-resolution matching algorithm. Categories are represented as sets of known instances. In this instance-based approach, storing and forgetting rules optimise memory usage. Classifier combinations are based on majority voting and the Dempster–Shafer evidence theory. 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. A simple physical 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. Various aspects of the approach are evaluated through systematic experiments.

Keywords: vocabulary acquisition; open-ended category learning; instance-based learning; long-term learning; computer vision; multi-resolution analysis; metacognition

1. Introduction

Human–robot interaction is currently a very active research field (Fong, Nourbakhsh and Dautenhahn 2003; Spexard, Hanheide and Sagerer 2007; Hegel et al. 2007). The role of social interaction in machine learning and, particularly, in robot learning is increasingly being 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

*Corresponding author. Email: lsl@ua.pt

ISSN 0954-0091 print/ISSN 1360-0494 online

DOI: 10.1080/09540090802413228
http://www.informaworld.com

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 behaviours. There is, therefore, a trend to develop robots with spoken language capabilities (Seabra Lopes and Connell 2001; Steels and Kaplan 2002; Seabra Lopes 2002; Fong et al. 2003).

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 2007a). Association and resemblance arise from perceptual, sensorimotor and functional aspects, whereas 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 connectionist/symbolic approach to the 'symbol grounding problem', which consisted of bottom-up grounding of symbolic representations in categorical representations (the invariant features of which would be learned through connectionism) and iconic representations.

Analogies with formal symbol systems and computer languages have led many to treat human language as a code, that is, a determinate set of tokens manipulated according to a determinate set of rules. By contrast, a distributed view on language origins, evolution and acquisition is emerging in linguistics. In this new perspective, language is treated as a cultural product, perpetually open-ended, incomplete and ambiguous to some extent (Love 2004). Rather than being an internal code, language is an external cognitive tool that simultaneously reflects cultural conceptualisations of the world and helps to create internal conceptualisations in individuals. It is thus recognised that 'language and cognitive dynamics are mutually constitutive' (Belpaeme and Cowley 2007).

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, 2003). 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 behaviour from a previous generation, or from an independent population (Kirby and Hurford 2002; Steels and Kaplan 2002). Given that language acquisition and evolution, in both human and artificial agents, involve not only internal, but also cultural, social and affective processes, the underlying mechanism has been called 'external' or 'extended' symbol grounding (Cowley 2007a; Belpaeme and Cowley 2007).

In general, the success of language acquisition in robots depends on a number of factors (Seabra Lopes and Chauhan 2007a): sensors; active sensing; physical interaction with objects; consideration of the affordance of objects; interaction with the human user; object and category representations; category learning; and category membership evaluation. Most of these issues still need to be suitably addressed by robotics researchers.

While keeping in mind the distributed/extended nature of language acquisition, this paper focuses on the required internal inference and memory processes for open-ended category learning and vocabulary acquisition. The emphasis on open-endedness is justified by two main factors. On one hand, human language is open-ended, as mentioned. So, it is not viable to pre-define a vocabulary and a corresponding set of categories. On the other hand, robots currently are rather limited in their perceptual and sensorimotor abilities, which prevents them from acquiring large subsets of a human language. In this context, open-ended category learning is necessary to support the adaptation of robots with limited abilities to specific users, tasks and environments.

As in other works reported in the literature, the vocabulary acquisition subject will be explored in a visual 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. As a general rule, the more imageable or concrete the referent of a word is, the easier it is to learn (Gillette, Gleitman, Gleitman and Lederer 1999). 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 series of words for naming certain categories of objects. Roy and Pentland (2002) presented 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 emphasise 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 categorisation. In a pre-linguistic phase, shape, colour 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 categorisation. A total of 12 object categories (pictures of animals in a book for small children) were learned in experiments.

The authors of this paper have previously developed a vocabulary acquisition and category learning system that integrates the user as instructor (Seabra Lopes and Chauhan 2007a). 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 six to 12 categories of regular office objects, associating them to natural language words. Like us, Lovett and co-workers (Lovett, Dehghani and Forbus 2007) also advocate that the key to recognition in the absence of domain expectations (i.e. in open-ended domains) is efficient online learning, but the work they describe is still based on the traditional procedures of gathering instances manually, training a recogniser on some of them and finally testing on unseen instances. The most notable feature of Lovett et al.'s work is the use of qualitative image representations and a specific similarity assessment method. The approach is demonstrated by learning eight categories of user-drawn sketches. Another recent work also explores continuous learning for visual concepts (Skocaj, Berginc, Ridge, Stimec and Hawes 2007). They used 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 and Honavar 2001) and EBNN (Thrun 1996). The instance-based learning system of Aha, Kibler and Albert (1991), although incremental, was demonstrated only on closed domains, i.e. domains with a pre-defined set of categories. In the only domain with more than 10 categories, actually a domain with 22 categories, best accuracy results were under 40%. 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; Cangelosi 2005). However, very few researchers have investigated agent architectures supporting open-ended category learning, symbol grounding and language acquisition.

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 accuracies achieved on this problem using 15 training images per category are 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 online learning capabilities.

In the context of classical symbolic artificial intelligence (AI), the issue of long-term learning remains largely an open issue. Attempts to apply classical AI learning techniques (explanation-based learning, case-based reasoning) in the long run have faced computational performance problems (Minton 1990; Mooney 1989; Francis and Ram 1993). The most common explanation for such problems is related to the cost of testing the applicability of the acquired knowledge (production rules, cases, etc.) to concrete problems. As problem-solving time increases with the number of learned structures, there is a trade-off between utility and cost. A recent study investigated the support provided to long-term learning by two well-known cognitive architectures, namely Soar and ACT-R (Kennedy and Trafton 2007). They reported computational performance problems in both systems, namely an increase in problem-solving time, as learning continues. In the case of ACT-R, analysed in more detail, one of the problems also identified is the inability to handle smoothly a finite and limited memory capacity.

The trade-off between utility and space/time costs is also an issue in instance-based category learning systems, as used in the visual category learning work described in this paper. In this domain, however, discrimination capacity tends to be the main factor limiting the overall performance, including the number of learned categories. Seabra Lopes and Chauhan (2007a) assume that a long-term category learning process in an artificial agent will eventually reach a breakpoint, that is, an internal state of the agent in which new categories can no longer be discriminated.

In this paper, a learning architecture is presented that can be used by robotic agents for long-term and open-ended learning of visual categories. An instance-based approach, with simple feature spaces, is adopted for category representation. Adequate categorisation relies on two main ingredients: similarity assessment based on multi-resolution matching; and a metacognitive self-monitoring and control loop. 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 base classifiers are used to reconfigure dynamically 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.

The paper is organised as follows. Section 2 presents the category learning architecture. Section 3 presents the used representations as well as the memory management approach. Section 4 describes similarity measures and basic classifiers. Section 5 presents classifier combinations and how the output category is predicted. Section 6 describes the performed experiments as well as the obtained results. Finally, Section 7 presents the conclusions.

2. Category learning architecture

Language acquisition is highly dependent on the representations and methods used for category learning and recognition. Open-ended category learning and language acquisition must be supported by long-term learning capabilities. Long-term learning is usually included at the core of cognitive theories (Kennedy and Trafton 2007). Learning a human language will require the participation of the human user as teacher or mediator (Steels and Kaplan 2002; Seabra Lopes and Chauhan 2007a).

To support the experimental work in our project, a simple agent was developed. It consists of 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 the 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.

A learning system in a robot should support long-term learning and adaptation. In the context of language acquisition, such a system should support supervised, incremental, online, opportunistic and concurrent learning and should also be able to improve or optimise its performance through meta-learning (Seabra Lopes and Wang 2002; Seabra Lopes and Chauhan 2007a).

The learning architecture proposed here (see Figure 1) was designed to satisfy these requirements. By organising its categories and instances according to user's feedback, it behaves in a supervised way. It is online 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 be given in a training phase or according to a predefined training schedule. It does not involve heavy computations, which facilitates the concurrent handling of multiple learning problems.

The proposed architecture includes a significant metacognitive component. The considered metacognitive capabilities are concerned with dynamic reconfiguration/control of the categorisation system as well as memory management. The term 'metacognition' was coined in psychology to refer to the phenomenon of cognition about cognition (Flavell 1971). The general information-processing framework of Nelson and Narens (1990) is well known for its explicit consideration of metacognitive processes. Basically, this framework divides cognitive processes into object-level processes and meta-level processes. A so-called *monitoring flow of information* from the object-level to the meta-level allows the meta-level to keep a dynamic model of the object-level. Based on this model, the meta-level can send a *control flow of information* to the object-level, determining which object-level behaviours are initiated, modified or terminated.

Researchers have recently been moving towards the conclusion that human category learning relies on multiple memory systems and multiple representations (Ashby and Obrien 2005; Kruschke 2005). Attentional selection, i.e. a mechanism of focusing on specific features or

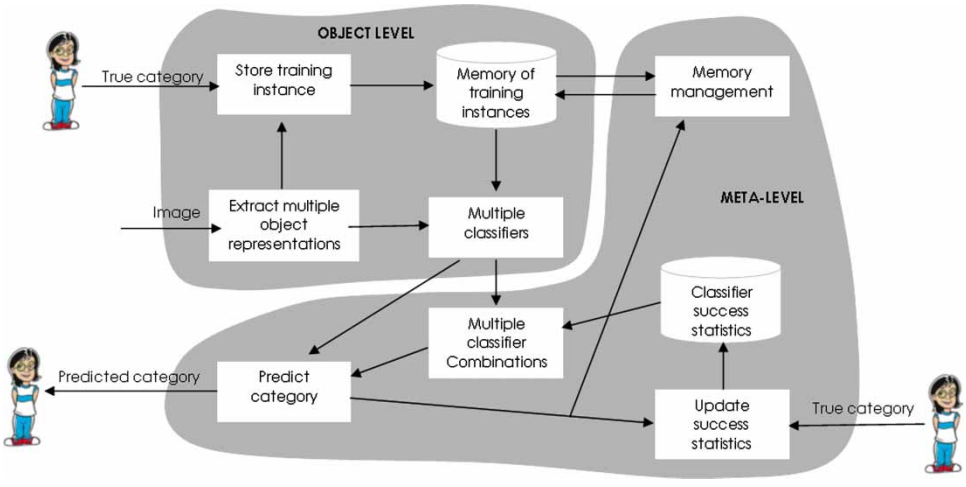

Figure 1. Category learning architecture.

representations based on recent experience, has also recently been emphasised (Kruschke 2005). These developments too suggest there is a meta-level associated to human category learning.

A lot of work in the AI field can also be seen as the interaction between object-level and meta-level processes. Cox (2005) presents a detailed survey of metacognition, in both psychology and artificial intelligence, and with the emphasis on problem-solving and story understanding tasks. Although some caveats are pointed out, metacognition is here considered an essential component for systems addressing such tasks.

The learning architecture presented in this paper (Figure 1) 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, Krzyzak and Suen 1992). The main innovation in this architecture is that those complementarities are explored in an online learning architecture, and a simple form of meta-learning takes advantage of the online 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 reconfigure dynamically some of the classifier combinations, and may also support the selection of the classifier that will predict the category of a new unseen object. Thus, the whole system is clearly divided into two main components. The object-level component extracts object representations, stores instances in memory and runs basic classifiers. The meta-level component optimises memory usage, monitors classifier performance, configures and runs classifier combinations and produces the category prediction for the target object.

3. Representations and memory

When the user points the mouse to an object in the scene image and selects an action to take (*teach*, *ask*, *correct*), an edge-based counterpart of the whole image is generated (Figure 2). From this edges image, the boundary of the object is extracted taking into account the user pointed position, and assuming that different objects do not occlude each other in the image. Given the boundary of the object, an edges-based image of the object is extracted from the full scene image. Finally, from the edges image of the object, different object representations are extracted and eventually

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

stored in memory. A metacognitive memory management process attempts to maximise storage utility while minimising categorisation costs.

3.1. *Extracting multiple object representations*

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

In the approach of this paper, multiple, possibly complementary feature spaces are explored concurrently. Most of these feature spaces are the result of segmenting the smallest circle enclosing the edges image of the object and centred in its geometric centre. For different feature spaces, such a circle is segmented either into a number of slices (Figure 3, left) or a number of concentric layers (Figure 3, 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:

- *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 4a for the three objects shown in Figure 2.
- *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-dependent feature space used in this work.
- *Shape slices normalised radii averages (SSNRA).* For each slice, i, the average radius of all edge pixels in that slice, R_i, is computed. In this feature space, an object is represented by a vector $\bar{\mathbf{r}} = r_1 \ldots r_{40}$, 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 2007a). An example is given in Figure 4b for the three objects shown in Figure 2.
- *Normalised radius standard deviation (RADSD).* This is another feature space composed of a single feature. Its value is the standard deviation of the normalised radii averages, $\bar{\mathbf{r}} = r_1 \ldots r_{40}$, mentioned in the previous paragraph.
- *Shape slices normalised 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 $\bar{\mathbf{s}} = s_1 \ldots s_{40}$, where $s_i = S_i/R$ and R is the average radius as mentioned above. An example is given in Figure 4c for the three objects shown in Figure 2.
- *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 both scale-invariant and rotation-invariant. An example is given in Figure 4d for the three objects shown in Figure 2.
- *Colour ranges (COLOR).* In this feature space, an object is represented by a set of the main colours of the object. Each colour is represented as a range of hue values in HSV colour space. These colour ranges are extracted from a colour histogram using a simple method presented in a previous paper (Seabra Lopes, Chauhan and Silva 2007). The primary purpose of using HSV format lies in the fact that most of the colour information is in the hue component (hue specifies the dominant wavelength of the colour in most of its range of values), thus facilitating image analysis based on a single colour dimension.

In summary, the current system uses two uni-dimensional feature spaces (*AREA* and *RADSD*), one colour-based feature space (*COLOR*) and four shape-based feature spaces (*SSH*, *SSNRA*, *SSNRSD* and *SLH*). Size information is captured by *AREA*. All other feature spaces are scale-invariant. All feature spaces are also rotation-invariant, except those based on slices (*SSH*, *SSNRA*

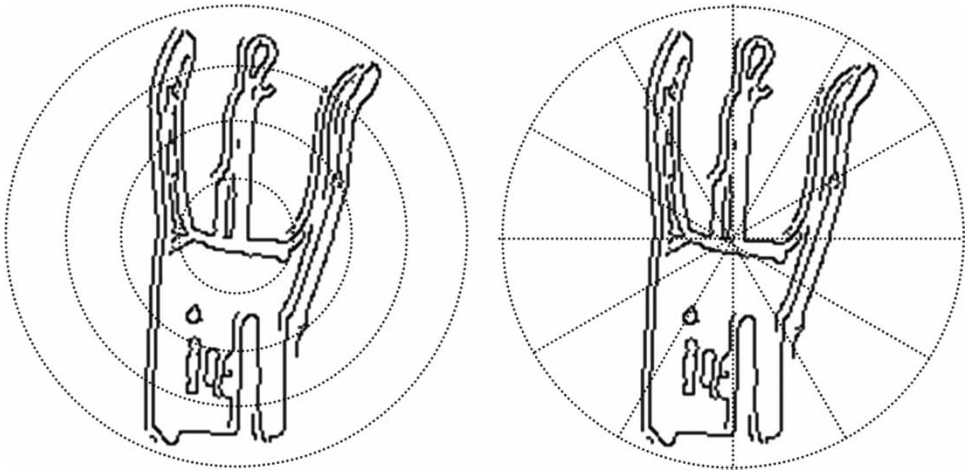

Figure 3. Segmentation of edges image of an object into (left) slices and (right) layers.

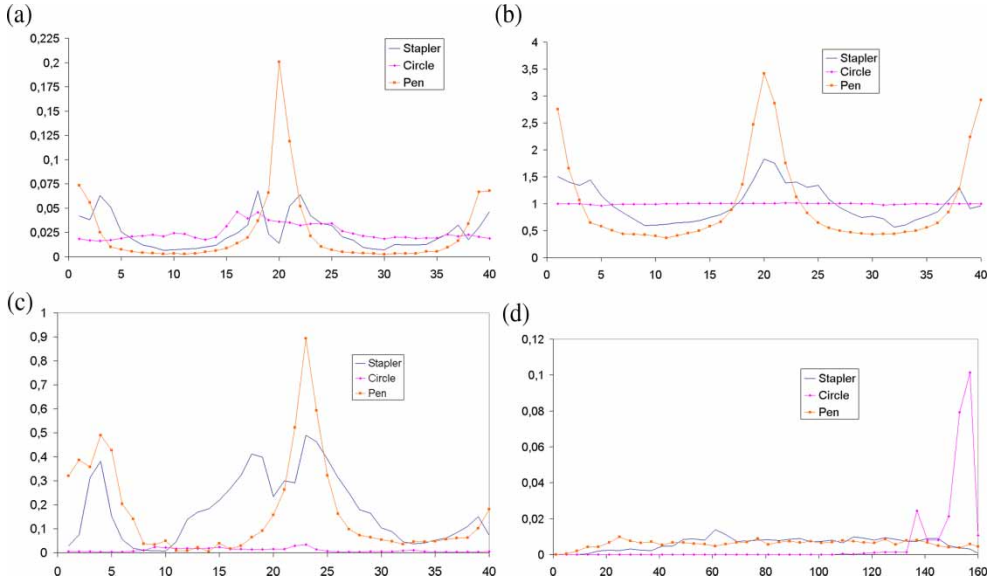

Figure 4. Different types of shape feature for the three objects in Figure 2: (a) shape slices histogram; (b) shape slices normalised radii averages; (c) shape slices normalised radii standard deviations; and (d) shape layers histogram.

and *SSNRSD*). For slice-based feature spaces, similarity can still be computed in a rotation-invariant way. Specifically, similarity (or distance) between any two instances is computed as the maximum similarity (or minimum distance) between the respective feature vectors as they are circularly rotated relative to each other.

The shape representations used here can be related to the so-called 'shape context' of Belongie, Malik and Puzicha (2002). In fact, their log-polar bins are a division of the object into slices and layers. However, the shape context is not centred in the geometric centre of the object. Instead, a shape context is computed for every edge pixel of the object. Object matching is then carried out by pairing edge pixels such that the total matching cost between histograms is minimised. So, the shape context is used as a local descriptor, whereas our feature spaces based on slices and layers are used for global object representation. Another important difference is concerned with

computational complexity. While most of the steps of the similarity computation algorithm of Belongie et al. (2002) run in time quadratic to cubic in the number of edge pixels, our representations can be built and used in linear time. Another state-of-the-art method in computer vision and object recognition is SIFT (Scale-invariant Feature Transform) (Lowe 2004). As we observe, SIFT performs quite well in specific categories (e.g. bottle of a specific brand of beer), but tends to perform poorly with general categories (e.g. any kind of bottle). Moreover, 'bag of feature' approaches such as SIFT provide little structural information about objects, which can be seen as a disadvantage from the point of view of language acquisition (Roy and Reiter 2005).

3.2. *Memory management*

An instance-based approach is adopted for category representation in which categories are simply represented by sets of known instances. New instances are stored in the following situations:

- If the user explicitly teaches the category of a given object through the *teach* action, the object representation is stored as an instance of the category.
- After an incorrect agent's prediction of the category of a given object, if the user provides the true category of the object (*corrective feedback*), then the agent adds the object to the set of instances of the category.

The idea is that, in the normal activity of a goal-directed learning agent, most instances are stored after corrective feedback, whereas the objects for which the agent predicts the correct category are not stored at all.

Meanwhile, as pointed out in the Introduction, there is a trade-off between the utility of storing an instance, on one side, and the increase in memory consumption and category membership evaluation costs, on the other. Also, since we focus on long-term learning in an open-ended domain, it must be noted that the instance-based representation of a given category may need to vary in time, as the set of known categories is expanded. In this process, some stored instances, which may have been sufficiently representative at a given point, may become redundant and/or useless (or even misleading) at a later stage. Instances typically become redundant when other instances of the same category are added closer to the category's boundary. A natural way of handling these problems is to develop memory management procedures that result in some form of forgetting.

As long-term learning in artificial agents remains largely an open issue, the same happens with forgetting strategies. Kennedy and Trafton (2007) emphasise that most cognitive systems do not explicitly forget learned knowledge. Forgetting has been addressed for speed-up learning, case-based reasoning and instance-based learning systems. Markovitch and Scott (1988) have shown that random forgetting of up to 90% of learned productions (macro-operators) in a problem-solving domain can improve global performance. Kennedy and De Jong (2003) extended the Soar cognitive architecture to remove productions that have not been used for more than some time, achieving statistically better computational performance. Francis and Ram (1993) include case deletion as one of the possibilities for coping with swamping problems in case-based reasoning systems. Mensink and Raaijmakes (1988) implement forgetting by a temporal decrease in the probability of retrieving an item from memory. Montaner, López and de la Rosa (2002) keep a measure of the importance of each memory item, which is strengthened when retrieved and is otherwise decayed in time. Finally, in the incremental instance-based learning approach of Aha et al. (1991), instances are stored only if misclassified and may be deleted eventually if they are considered noisy. The classification accuracy of individual instances stored in memory is evaluated as new instances are introduced. Based on this information, the stored instances that are believed to be noisy are discarded.

For the category learning system of this paper, a single forgetting rule is implemented. The rule for forgetting is basically the same as the rule for remembering. As mentioned, an instance is stored in memory if it cannot be classified correctly. So, in abstract terms, the rule for forgetting is the following:

- An instance can be removed from memory (forgotten) if it will still be classified correctly after removal.

This rule is applied conservatively. Each time the system fails to classify correctly a new instance, leading to addition of the instance to the database, the metacognitive component of the system will check whether any other instance of the same category can be removed. As soon as one instance satisfying the above rule is found it is removed, and the remaining instances are kept in memory. Only on addition of another instance will the other existing instances be considered for removal.

4. Basic inference and categorisation

Categorisation 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.

4.1. *Euclidean membership measure*

As objects are represented as feature vectors in most of the feature 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 normalising 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, $i, k = 1, \ldots, N$, 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.

4.2. *Multi-resolution membership measure*

In this 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, $w_i = 12^i$ 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 that 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 (2007), but also to other feature vectors that are normalised by some constant R, as happens in the 'shape slices normalised 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 over a base-level ordered feature vector (Seabra Lopes and Chauhan 2007a; 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, $i, k = 1, \ldots, N$, 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.

4.3. *Base classifiers*

Categorising 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.

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*); and 'normalised radius standard deviation' (*RADSD*).
- Classifiers using feature vectors with Euclidean membership measurement: 'shape slices histogram' (*SSH-EM*); 'shape slices normalised radii averages' (*SSNRA-EM*); 'shape slices normalised 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 colour-based category representation and membership measure (*COLOR*) presented elsewhere (Seabra Lopes et al. 2007).

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

5. Classifier combinations and meta-learning

The complete learning and categorisation approach includes classifier combinations. Some of the classifier combinations are dynamically reconfigured according to the observed success of the base classifiers. This introduces a meta-learning component in the category learning system.

5.1. *Classifier success*

The meta-level component of the agent's architecture (see Figure 1) is responsible for maintaining updated success statistics for all classifiers. Each time the agent sees an object and the user provides its category, the agent runs all classifiers on the object and compares the results with the user-provided category. Then, for each classifier, the respective success measure is updated as follows:

$$S_t = wS_{t-1} + (1 - w)R_t,$$

where t identifies a teaching iteration (in which the user teaches, validates or corrects the category of a given instance), R_t is the result of the classifier in the tth iteration ($R_t = 1$ if correct category, $R_t = 0$ otherwise) and S_t is the updated measure of success of the classifier in the tth iteration, computed as a weighted average. The parameter w is the weight of the previous value of the success measure, S_{t-1}. The weight parameter is computed with reference to a window of a certain number of iterations, W. In the initial phase, while $t \leq W$, the weight is computed as $w = (t - 1)/t$, which results in a success measure equal to the arithmetic average of all results R_i so far ($i = 1 \ldots t$). For the general case of $t > W$, the weight is computed as $w = (W - 1)/W$, which results in gradual forgetting of older results to reflect the most recent performance.[1]

5.2. *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 one to derive multiple basic belief assignments, 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 normalised 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 normalised membership measures are used instead of conditional probabilities. These conditional probabilities could be estimated based on the confusion matrices 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. In such a scenario, therefore, the alternative would be to build the confusion matrices online. 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 did not seem to be able to outperform membership measures significantly, 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*). As 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.

5.3. *Majority voting combinations*

Voting methods are also well known in classifier combinations (Xu et al. 1992; Kittler, Duin and Matas 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.

5.4. *Predicting the category of the target object*

The internal computations described up to 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.

6. Experimental evaluation

6.1. *Teaching protocol for experimental evaluation*

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 need to scale up has been pointed out by several authors (Cangelosi and Harnad 2000; Steels and Kaplan 2002; Cangelosi 2005). 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 measure 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 reasonable value.

Robots and software agents, however, 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 Algorithm 1 was proposed by Seabra Lopes and Chauhan (2007a). For clarity, its presentation is repeated here.

This protocol is applicable for any open-ended category learning domain. For every new category the instructor introduces, the average precision of the whole system is calculated by performing classification with all known categories. To that end, the instructor repeatedly shows instances of the known categories, checks the agent's predictions and sends corrections when necessary (this is referred below as a 'question/correction iteration'). Average precision is calculated over the last $3 \times n$ classification results (n being the number of categories that have already been introduced). The precision of a single classification is either one (correct category) or zero (wrong category). When the number of classification results since the last time a new category 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 category is based on the precision threshold. In all experiments described below, this threshold was set to 0.67, as in previously published work.

It should be noted that classification precision, as described in this section, is an external measure that controls the application of the teaching protocol for experimental evaluation. Therefore, it should not be confused with the classifier success measure, internally computed by the agent and used to drive classifier combinations, as presented in Section 5.1. The main difference between both is that, whereas classifier success takes into account all classifier results since the agent was running (giving more weight to more recent results), the protocol's precision measure ignores any results given by the agent before the last time a new category was introduced. This is because the precision measure is used to analyse the impact of the introduction of a new category, from initial instability to recovery.

ALGORITHM 1 Teaching protocol for experimental evaluation.

```
introduce Category_0;
n = 1;
repeat {
        introduce Category_n;
        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).
```

6.2. *Long-duration experiment*

A long-duration experiment was conducted according to this protocol (Seabra Lopes and Chauhan 2007b). 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 experiment continued for 3767 question/correction iterations and took more than a week to complete (Figure 5). In total, it was possible to teach 68 categories of real-world objects, which can be roughly grouped as follows: 40% are office objects; 20% are child toys; 20% are other home objects; and the remaining 20% are objects of varied types. Some of the categories were represented by several objects, whereas others were represented by a single object. Figure 7 displays one sample image per category. During the teaching/learning process, the agent stored a total of 1168 training instances.

Figure 5 displays the evolution of classification precision versus number of question/correction iterations. Sections of the curve with more pronounced oscillations indicate the introduction of new categories. As observed in previous work (Seabra Lopes and Chauhan 2007a; 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 5, we see that most categories were learned in the first ~2000 iterations (exactly 60 categories), whereas in the remaining ~1800 iterations it was possible to learn only eight additional categories.

The breakpoint is also clearly visible in Figure 6, 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 four 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 nine instances per category. Finally, while learning the last eight categories, the number of instances per category abandoned the linear evolution trend and jumped to 17.

The analysis of the performance of the individual classifiers is also relevant (Table 1). 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 classification success 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%).

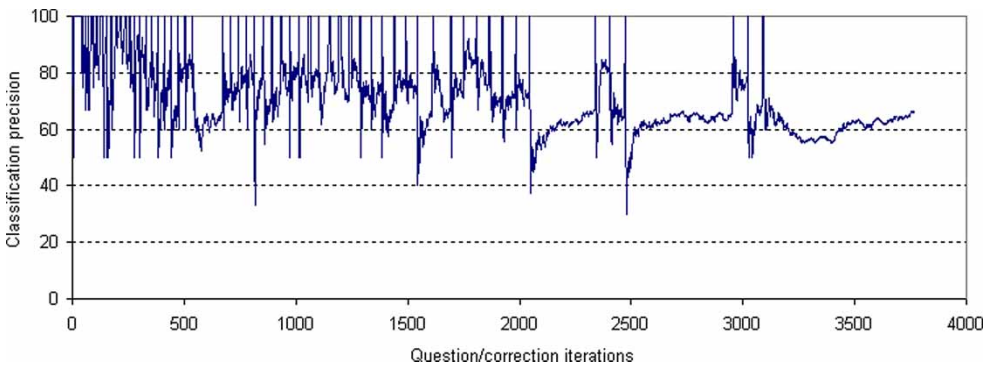

Figure 5. Evolution of classification precision versus number of question/correction iterations.

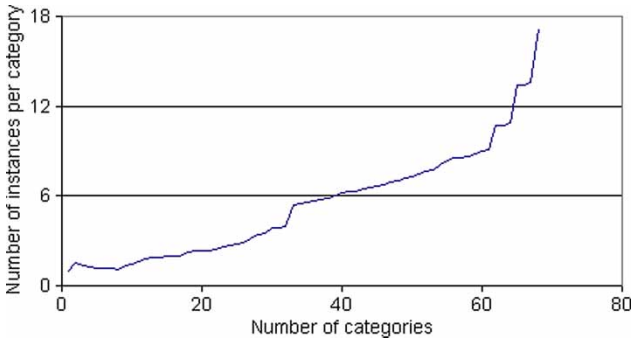

Figure 6. Evolution of number of training instances per category versus number of acquired categories.

Figure 7. Sample images of all acquired categories.

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

The externally observable performance of the agent was 70.0%, very close to *MAJORITY-ALL*. Note that, as mentioned before, the predictions of the agent are those of the current most successful

Table 1. Average 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
Colour-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

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%.

6.3. *Semi-automated experiments*

While the described long-duration experiment was carried out, all analysed images for which categories were provided by the user were saved along with the respective categories. This enables partially automated experiments to be performed. For that purpose, a simulated user was implemented, which interacts with the learning system following the above-mentioned protocol. The actions of the simulated user are the same as those of a human user: *teach*, *ask* and *correct*. The simulated user picks images randomly for interaction with the learning system and uses each stored image at most once. When the agent is ready to learn a new category, the next category is randomly selected. When the simulated user runs out of images for a particular category, the human user is called to interact with the agent, so that the experiment can continue with its normal course. Each time the simulated user calls the human user for interaction with the agent, additional images are collected and stored.

With this methodology, it is possible to conduct systematic experiments that would, otherwise, take many weeks to complete. These semi-automated experiments were conducted to evaluate different configurations of the learning system. The obtained results are given in Table 2. The values presented in the table are averages and standard deviations computed over five experiments in each particular configuration.

In the most successful individual experiments, the number of learned categories reached 55, still clearly lower than the 68 categories learned in the initial long-duration experiment. This is explained by two main factors. First of all, owing to contingencies in the image processing software, the images retrieved from the simulated user's pool occasionally produce edges different from the original images before storage. Second, when the simulated user called the human user

Table 2. Evaluation of several configurations of the category learning system.

Classifiers	Prediction method	Forgetting	No. of iterations	No. of categories	No. of instances/category
Only *SSNRA-PM*		No	3307 ± 1246	41.4 ± 5.4	28.9 ± 8.5
All classifiers except *MAJORITY-ALL*	Current best	No	3007 ± 1062	44.4 ± 4.2	24.2 ± 7.0
All classifiers except *MAJORITY-ALL*	Majority voting	No	2138 ± 340	44.2 ± 5.5	26.6 ± 1.8
All classifiers	Current best	No	3302 ± 730	49.4 ± 4.8	23.4 ± 3.6
All classifiers	Current best	Yes	3065 ± 529	45.2 ± 3.7	12.9 ± 1.8

Note: Values are averages and standard deviations computed from sets of five experiments.

for physical interaction, new objects were used, increasing the intra-category heterogeneity and making the categories harder to learn (a total of 109 objects of 68 categories were used in these experiments). These new conditions, however, do not affect the conclusions that will be drawn.

These experiments were designed to evaluate three main aspects: (1) Which prediction method works best? (2) What is the impact of metacognitive processing in learning performance? (3) What is the impact of the forgetting method in memory usage and learning performance?

Two meta-level prediction methods for predicting the output category of the target object were evaluated: (1) Use the currently most successful classifier; and (2) use the majority voting of all classifiers. The evaluation was done with all base classifiers and all dynamically reconfigured classifier combinations (*DS2TOP*, *DS3TOP*, *DS4TOP*, *DS5TOP*, *MAJ3TOP* and *MAJ5TOP*). The two methods enable learning similar numbers of categories (in both cases, 44 categories on average) with similar numbers of stored training instances (around 25 instances per category in breakpoint). However, majority voting seems to enable faster learning.

The configuration of the system used in the long-duration experiment included majority voting as one of the classifiers (*MAJORITY-ALL*) and then predicted the output category of the target object using the current best classifier. On the new experiments, this configuration enabled an average of 49 categories to be learnt with a standard deviation of 4.8. The number of stored instances per category is slightly lower than in the previous two cases.

To evaluate the impact of metacognitive processing in the number of learned categories, a comparison was done with the best individual classifier, namely, *SSNRA-PM* (shape slices normalised radii averages with pyramid matching). *SSNRA-PM* was able to learn 41 categories on average, with the highest number of stored instances per category. With respect to this result, the best multi-classifier configuration yielded an improvement of 19.3% on average (from 41.4 to 49.5 categories).

Finally, the forgetting method was evaluated by comparison with the configuration in which all classifiers are used and the current best classifier predicts the output category. The forgetting method enabled the reduction of memory consumption in 45% at the expense of losing 8.5% in number of learned categories. It is important to note that, while significantly improving memory consumption, the forgetting rule does not affect the general trend observed in Figure 6 for the evolution of number of stored instances per category versus number categories. So, when the learning process approaches breakpoint, the number of stored instances per category starts to grow faster and faster.

7. Conclusion

In order to interact and communicate with human users using natural language, adapting to different users, tasks and environments, robots need to acquire language from the users themselves.

The internal information processing abilities of robots must therefore support this vertical language transmission process. In particular, robots must be capable of long-term and open-ended category learning.

This paper has presented a learning architecture that can be used by robotic agents for long-term and open-ended category learning. To be more adaptive and to improve learning performance as well as memory usage, this learning architecture includes a metacognitive processing component. Multiple object representations and multiple classifiers and classifier combinations are used.

The shape-based object representations are an original proposal of the authors in an attempt to develop computationally light classifiers that can be used in an online classifier combination architecture. These shape representations are scale-invariant. One of them is also rotation-invariant and the others easily allow for rotation-invariant similarity assessment. The application of the pyramid matching algorithm of Grauman and Darrell (2007) to feature spaces where objects are described not by histograms but by other normalised 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%) in a long-duration experiment illustrate this point.

Classifier combinations are based on majority voting and the Dempster–Shafer evidence theory. 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 this work, dynamic reconfiguration of a classifier combination consists of modifying the set of basic classifiers that are included in the combination, giving preference to those with higher classification success. This attentional selection mechanism is important for the agent to adapt to specific domains, as it allows the agent to choose the classifier combinations that are more suited to the categories that it actually needs to discriminate. It therefore directly supports the open-ended nature of language acquisition. Although classifier combinations are common in off-line learning, its use in an online learning architecture is novel, as far as we know. In the experiments, classifier combinations enabled an average improvement in learning performance of 20%, measured as the increase in number of learned categories with respect to the best single classifier.

Categories are represented as sets of known instances. In this instance-based approach, storing and forgetting rules optimises memory usage. In experiments, the forgetting rule enabled a reduction of memory consumption of 45% on average, at the expense of 8.5% deterioration of learning performance, as measured by number of learned categories.

A simple physical agent, incorporating these learning capabilities, was used to test the approach. Overall, our approach seems to outperform several previous works initially cited, in which it was possible to learn up to a maximum of 12 categories. In a long-duration experiment, and with the help of a human mediator, our agent incrementally learned 68 categories of real-world objects visually perceivable through an inexpensive camera. The same configuration enabled between 45 and 55 categories to be learnt in slightly different conditions (more heterogeneous categories, minor difference in edge detection).

Although the paper focused on visual category learning, it would be interesting and relevant to extend the proposed learning architecture to other domains, including sensorimotor categories and phonetic categories. Sensorimotor categories can be learned through action, either self-performed or observed. Apart from action categories themselves, this is relevant for taking affordances into account when categorising objects. Phonetic categories are important when using spoken language. This poses additional problems and requires an integrated learning architecture that takes into account how learning object and action categories and learning phonetic categories bootstrap each other (Roy and Pentland 2002). Analogies with how human infants acquire language through

interaction with caregivers can provide insight for developing similar mechanisms in robots (Cowley 2007b).

Acknowledgements

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. We would like to thank Stephen J. Cowley, Michael Anderson and the three anonymous reviewers for many helpful comments.

Note

1. In the implementation, a window of $W = 50$ iterations and the corresponding weight $w = 0.98$ were used. With these parameters, the result is that the latest 20 iterations (i.e. iterations $t - 19$ to t) have a combined weight of approximately $1/3$ in the success value, the rest of the window (iterations $t - 49$ to $t - 20$) account for another $1/3$ of the success value and all other older iterations (1 to $t - 50$) account for the remaining $1/3$ of the success value.

References

Aha, D.W., Kibler, D., and Albert, M. (1991), 'Instance-based Learning Algorithms,' *Machine Learning*, 6, 37–66.

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 O'Brien, J.B. (2005), 'Category Learning and Multiple Memory Systems,' *Trends in Cognitive Science*, 9, 83–89.

Belongie, S., Malik, J., and Puzicha, J. (2002), 'Shape Matching and Object Recognition Using Shape Contexts,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 509–522.

Belpaeme, T., and Cowley, S.J. (2007), 'Extending Symbol Grounding,' *Interaction Studies*, 8, 1–6.

Cangelosi, A. (2005), 'Approaches to Grounding Symbols in Perceptual and Sensorimotor Categories,' in *Handbook of Categorization in Cognitive Science*, eds. H. Cohen and C. Lefebvre, Oxford: Elsevier Science, pp. 719–737.

Cangelosi, A., and Harnad, S. (2000), 'The Adaptive Advantage of Symbolic Theft over Sensorimotor Toil: Grounding Language in Perceptual Categories,' *Evolution of Communication*, 4, 117–142.

Cowley, S.J. (2007a), 'Distributed Language: Biomechanics, Functions and the Origins of Talk,' in *Emergence of Communication and Language*, eds. C. Lyon, C. Nehaniv and A. Cangelosi, London, UK: Springer, pp. 105–127.

Cowley, S.J. (2007b), 'How Human Infants Deal With Symbol Grounding,' *Interaction Studies*, 8, 83–104.

Cox, M.T. (2005), 'Metacognition in Computation: A Selected Research Review,' *Artificial Intelligence*, 169, 104–141.

Flavell, J.H. (1971), 'First Discussant's Comments: What is Memory Development the Development of?,' *Human Development*, 14, 272–278.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003), 'A Survey of Socially Interactive Robots: Concepts, Design, and Applications,' *Robotics and Autonomous Systems*, 42, 143–166.

Francis, A., and Ram, A. (1993), 'The Utility Problem in Case-based Reasoning,' in *Case Based Reasoning: Papers from the 1993 Workshop*, Washington, DC: AAAI Press, pp. 160–167.

Gillette, J., Gleitman, H., Gleitman, L., and Lederer, A. (1999), 'Human Simulations of Vocabulary Learning,' *Cognition*, 73, 135–176.

Grauman, K., and Darrell, T. (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.

Hegel, F., Lohse, M., Swadzba, A., Wachsmuth, S., Rohlfing, K., and Wrede, B. (2007), 'Classes of Applications for Social Robots: a User Study,' in *Proceedings of the International Symposium on Robot and Human Interactive Communication*, pp. 938–943.

Kennedy, W.G., and De Jong, K. (2003), 'Characteristics of Long-term Learning in SOAR and its Application to the Utility Problem,' in *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 337–344.

Kennedy, W.G., and Trafton, J.G. (2007), 'Long-term Symbolic Learning,' *Cognitive Systems Research*, 8, 237–247.

Kirby, S., and Hurford, J. (2002), 'The Emergence of Linguistic Structure: An overview of the Iterated Learning Model,' in *Simulating the Evolution of Language*, eds. A. Cangelosi and D. Parisi, Berlin: Springer, pp. 121–148.

Kitler, J., Duin, R.P.W., and Matas, J. (1998), 'On Combining Classifiers,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 226–239.

Kruschke, J.K. (2005), 'Category Learning,' in *The Handbook of Cognition*, eds. K. Lamberts and R.L. Goldstone, chap. 7, London, UK: Sage Publications, pp. 183–201.

Love, N. (2004), 'Cognition and the Language Myth,' *Language Sciences*, 26, 525–544.

Lovett, A., Dehghani, M., and Forbus, K. (2007), 'Incremental Learning of Perceptual Categories for Open-domain Sketch Recognition,' in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 447–452.

Lowe, D.G. (2004), 'Distinctive Image Features from Scale-invariant Keypoints,' *International Journal of Computer Vision*, 60, 91–110.

Markovitch, S., and Scott, P.D. (1988), 'The Role of Forgetting in Learning,' in *Proceedings of the Fifth International Conference on Machine Learning*, pp. 459–465.

Mensink, G., and Raaijmakers, J.G. (1988), 'A Model for Interference and Forgetting,' *Psychological Review*, 95, 434–455.

Minton, S. (1990), 'Quantitative Results Concerning the Utility of Explanation-based Learning,' *Artificial Intelligence*, 42, 363–391.

Montaner, M., López, B., and de la Rosa, J.L. (2002), 'Improving Case Representation and Case Base Maintenance in Recommender Agents,' in *European Conference, ECCBR 2002*, Aberdeen, Scotland, *Proceedings, Lecture Notes in Computer Science*, *No. 2416*, Berlin: Springer, pp. 234–248.

Mooney, R. (1989), 'The Effect of Rule Use on the Utility of Explanation-based Learning,' *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 725–730.

Nelson, T.O., and Narens, L. (1990), 'Metamemory: A Theoretical Framework and New Findings,' *The Psychology of Learning and Motivation*, 26, 125–173; reprinted in Nelson, T.O. (ed.) (1992), *Metacognition: Core Readings*, Boston: Allyn and Bacon, pp. 9–24.

Polikar, R., Udpa, L., Udpa, S.S., and Honavar, V. (2001), 'Learn++: An Incremental Learning Algorithm for Supervised Neural Networks,' *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 31, 497–508.

Roy, D., and Pentland, A. (2002), 'Learning Words from Sights And Sounds: A Computational Model,' *Cognitive Science*, 26, 113–146.

Roy, D., and Reiter, E. (2005), 'Connecting Language to the World,' *Artificial Intelligence*, 167, 1–12.

Seabra Lopes, L. (2002), 'Carl: From Situated Activity to Language Level Interaction and Learning,' in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 890–896.

Seabra Lopes, L., and Camarinha-Matos, L.M. (1998) Feature Transformation Strategies for a Robot Learning Problem, in *Feature Extraction, Construction and Selection. A Data Mining Perspective*, eds. H. Liu and H. Motoda, Dordrecht: Kluwer Academic, pp. 375–391.

Seabra Lopes, L., and Chauhan, A. (2007a), 'How Many Words Can My Robot Learn? An Approach and Experiments with One-class Learning,' *Interaction Studies*, 8, 53–81.

Seabra Lopes, L., and Chauhan, A. (2007b), 'Scaling-up Category Learning for Language Acquisition in Human–Robot Interaction,' in *Symposium on Language and Robots: Proceedings of the Symposium*, pp. 83–92.

Seabra Lopes, L., Chauhan, A., and Silva, J. (2007), 'Towards Long-term Visual Learning of Object Categories in Human–Robot Interaction,' in *New Trends in Artificial Intelligence*, eds. J.C. Maia Neves, M.F. Santos and J.M. Machado, Associação Portuguesa para a Inteligência Artificial, pp. 623–634.

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

Seabra Lopes, L. and Wang, Q.H. (2002), 'Towards Grounded Human–Robot Communication,' in *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 312–318.

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

Skocaj, D., Berginc, G., Ridge, B., Stimec, A., and Hawes, N. (2007), 'A System for Continuous Learning of Visual Concepts,' in *Proceedings of the International Conference on Computer Vision Systems*, Bielefeld, Germany.

Spexard, T.P., Hanheide, M., and Sagerer, G. (2007), 'Human-oriented Interaction With an Anthropomorphic Robot,' *IEEE Transactions on Robotics*, 23, 852–862.

Steels, L. (2001), 'Language Games for Autonomous Robots,' *IEEE Intelligent Systems*, 16, 16–22.

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

Steels, L., and Kaplan, F. (2002), 'AIBO's First Words: The Social Learning of Language and Meaning,' *Evolution of Communication*, 4, 3–32.

Thomaz, A.L., and Breazeal, C. (2006), 'Transparency and Socially Guided Machine Learning,' in *Proceedings of the 5th International Conference on Developmental Learning*.

Thrun, S. (1996), *Explanation-based Neural Network Learning: A Lifelong Learning Approach*, Boston, MA: Kluwer.

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

Yu, C. (2005), 'The Emergence of Links Between Lexical Acquisition and Object Categorization: A Computational Study,' *Connection Science*, 17, 381–397.