Active Object Discovery for Communicating with Humans

Claire D'Este and Claude Sammut
ARC Centre for Autonomous Systems
School of Computer Science and Engineering
University of New South Wales
Sydney, NSW 2052 Australia
{claired,claude}@cse.unsw.edu.au

Abstract

This paper describes the first layer of a system for human-robot communication. The goal is a robot that can learn about objects from many different teachers, who require many different levels of detail to describe them. The resulting system consists of robot sensing, actuating and machine learning methods that keep as flexible as possible the kinds of objects and properties that can be learned.

1 Introduction

Most robotics researchers are resigned to the fact that humans and robots have an entirely different experience of the world. They therefore create models of the environment that are entirely from the robot's point of view and that make use of its strengths. However, as the kinds of people using robots changes, to include areas like aged-care and construction, it will be very difficult for ordinary people to describe objects in terms of laser or sonar scans, or complex physics models. What these kinds of robot owners require is the ability to teach the robot about new objects using their own terms, and to decide to what level of detail they should be described. It would also be infeasible to have them teach the robot everything about each object from scratch. We would like the robot to be finding general features in objects that can be used to describe novel things. Once this was learned, even if the robot had never heard of an apple before, we could still ask it to fetch the green, round thing in the fruit bowl, and hopefully receive back something sane.

Within this work, the teacher presents an object for the robot to learn from, and chooses a feature of the object to describe, either colour, shape, size or weight. They then name the current value of this feature, through statements such as "This is brown in colour". The teacher can decide to specify as many values for colour as they like. If they only want to be able to distinguish brown from green, then this should be possible, or if they want to tell brown from beige, tan, or bronze, then the learning system should be able to accommodate.

The task also involves testing the robot on novel objects. We require the robot to learn general features from the objects that have been directly shown and labelled by the teacher, and apply them to objects it has never seen before. We should be able to teach the robot about round shapes from balls, and then show the robot a round plum and have it successfully answer "What shape is this?".

To complete this task we need to keep the types of discriminations and objects open as possible. So ideally, we need to make as few assumptions about the objects as possible. The kinds of objects the robot can learn about is severely limited by assuming that the object will be a bright and uniform colour (or texture, or shape etc), the only object of this colour, or the only object in a very specific environment. Instead of using these methods our system segments the object using motion.

For a open-ended classification system like this the number of categories cannot be specified beforehand, as with many techniques such as neural networks or other clustering algorithms. The learning method needs also to deal with exceptions and not lose old knowledge when new knowledge is learned. We attempted to use classification systems such as x-means [Pelleg and Moore, 2000]. However, although this does not require you to know the number of classes beforehand, you must chose the number of splits to divide up the input space, and the results are highly dependant on your choice. In order to make this choice you need to have a good idea about the number of clusters you want beforehand, and this is unlikely when it comes to vocabulary learning. Instead, we learn finer and finer classifications as it becomes necessary to do so. A technique that fulfils these requirements, and is therefore implemented in this system, is Ripple Down Rules.

This paper describes robot sensing, actuating and machine learning methods that allow the robot to learn different ways of describing objects from different human teachers.

1.1 Related Work

The Talking Heads experiments [Steels and Kaplan, 2002] involved software agents with pan-tilt cameras negotiating the meaning of words. The agents spoke about geometric shapes on a blackboard and the words to describe them mapped directly to predefined intervals. For example, the word 'gubo' could map to a predefined interval [LARGE], which was a scaled value greater than 0.5 and less than 1. This meant the software agents could not create their own feature categories, but were limited to mapping words to those categories designed by the programmer.

[Fitzpatrick, 2003a]'s robot tried to infer the names of objects from a guided activity with a human teacher. The robot is asked to find an object 'toma'. It is shown several objects and told yes or no, if this object is a 'toma'. The robot can later be shown the 'toma' object and successfully name the object. This system collects images of the object and then clusters them based on colour histograms. This is only successful for specific objects and can not be used to find general features for describing novel objects.

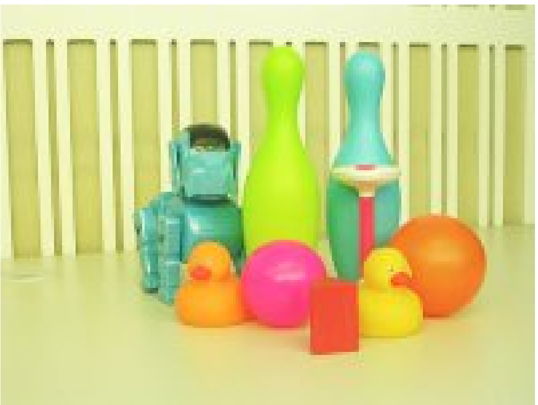

Figure 1: The kinds of objects the robot must learn from and about.

2 Implementation Details

2.1 Platform

The robot is a Sony AIBO ERS-7. This robot has a 64-bit RISC Processor, clock speed of 576 MHz and 64 MB of RAM. It has a total of twenty degrees of freedom, wireless LAN, stereo microphones, nine touch sensors, two infrared distance sensors, acceleration and vibration sensors and a colour camera with a resolution of 350,000 pixels. The images are grabbed using its on-board camera and the speech synthesis is transmitted with its own speaker. The speech recognition and vision processing is done off board. The speech recognition uses IBM ViaVoice, which sends sentences to be parsed by a pattern matching Prolog program.

2.2 The Objects

The objects (Figure 1) are mostly children's toys designed for children between one and three years old.

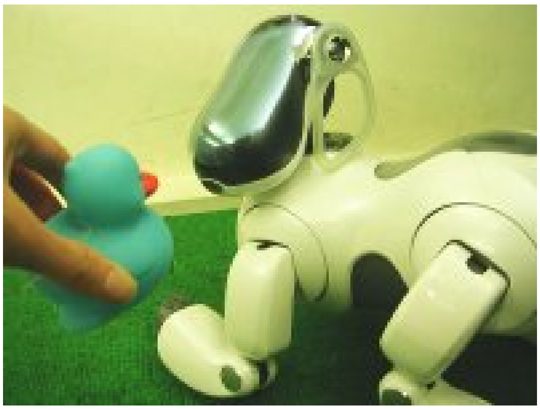

Figure 2: Waving an object before the robot so it can perform motion segmentation.

2.3 Object Segmentation

To keep the kinds of objects we can learn about as open as possible, we use the object's motion to segment it. This motion is initiated either by the teacher waving the object around (Figure 2), or pawing or prodding from the robot. One object, a toy gorilla, produces its own motion. The objects are discriminated from the background using a similar method to that described in [Fitzpatrick, 2003a] using the maximum flow and minimum cut algorithm from [Boykov and Kolmogorov, 2001]. The robot stays in one place and two images are taken in quick succession. Motion is presumed where the pixels have changed from image to image. A similar method might be used with stereo images, unfortunately the AIBO is equipped with a single camera. The algorithm then tries to cut around the changed pixels using cuts with the minimum cost.

Once we have a cut-out of the object as seen in Figure 3, we can then process the image using known computer vision algorithms. [Fitzpatrick, 2003a] uses a model-based approach to recognise specific instances of the object, as mentioned above. As we are hoping to learn more general concepts from instances, we analyse the image to discover generic information.

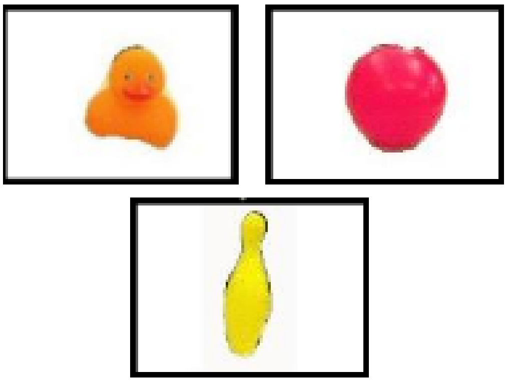

Figure 3: Examples of minimum cut/maximum flow from motion segmentation and colour classification of a duck, ball and skittle.

2.4 Ripple Down Rules

Ripple Down Rules (RDR) are a method for creating classification rules incrementally. This was developed in response to [Compton and Jansen, 1988]'s experience creating expert systems. The experts they studied, appeared to create rules that justified their final decision, rather than reflected their actual decision process. Ripple Down Rules also allow for recovery by adding exception rules. New rules are added when a classification error is made. If a rule exists that covers the current input, but requires different output, the system requests or searches for new features that will help to differentiate this exception to the rule.

Figure 4 shows an example of a Ripple Down Rule created for classifying colour. The rules can be read as:

```
if $x$ lies between [y-z] then $a$
        unless $x$ lies between [y-z] then $b$
else if $x$ lies between [y-z] then $c$
```

Alternative rules are printed on separate lines and indented rules are exceptions to the rule above [Scheffer, 1996].

Ripple Down Rules have been used to create rules to analyse medical images for lung boundary extraction [Misra, Sowmya, and Compton, 2004] and robot soccer scenes for a robot [Kerr and Compton, 2003]. In [Misra, Sowmya, and Compton, 2004], a medical professional utilises a graphical interface to adjust parameters such as pixel intensities and texture properties to allow them to customise the extraction algorithm themselves. In [Kerr and Compton, 2003] features such as colour and texture are determined outside of the RDR. The expert then hand-crafts recognition rules about objects in the soccer environment. For example, a ball is orange and should not be seen above the goal.

Although this system uses these expert system or knowledge acquisition techniques, the expert does not construct the rules. The expert only provides the correct classification, or knowledge that the object has been misclassified.

2.5 Features

Colour

The colour features are described by intervals in YUV (luminance, blue component, red component) space. Ripple Down Rules are created using intervals of YUV pixels found in the objects, and colour words given by the teacher. The maximum-flow/minimum-cut algorithm is also used for to extend intervals for the colour classification rules. All pixels in the image are classified using the current rules. Pixels with known colours are used as the foreground, as pixels that had moved were used in the motion segmentation. The pixel values within the cut region provide a range of shades and brightnesses of the colour and create a more robust classification of each colour. Figure 4 shows the RDR learned from the interaction in Table 1.

```
[0 .. 255][0 .. 255][0 .. 255] → 0
        [123 .. 205][34 .. 91][139 .. 156] → 1
               [123 .. 205][78 .. 91][139 .. 156] → 2
        [36 .. 210][63 .. 119][127 .. 181] → 2
0 = unknown, 1 = yellow, 2 = orange,
```

Figure 4: A learned RDR for classifying two colours.

Classifying colour using Ripple Down Rules is based on a system used for the University of New South Wales Robocup team [Sammut, 2005]. It is used to calibrate colours in a new environment under new lighting conditions. In this situation the robot takes images and a human uses a mouse to highlight the colours of the ball, goals and localisation-beacons.

Size

One of the aims of this project was to explore the additional information that could be obtained from having a robot that could walk around in the environment and interact with objects. Therefore, although other techniques could be used for determining the size of objects, such as head scanning and triangulation, we preferred to use methods which had a more interactive quality, or used its particular sensors. For determining the size of an object the robot walks backwards or forwards until it can get an accurate infrared reading of the distance d to the object. If the object is too close the robot reaches out and tries to touch the object. The distance is then measured by the amount the robot had to reach out before its touch sensor was triggered.

As the robot is moving around in a three-dimensional environment, it is important to discriminate between an object that appears small because it is small, or an object that is large and simply far away. The equation 1 is used to determine the actual size of the object o, where v is half the viewing angle, w is half the pixel width of the image, and a the total pixel area of the object.

$$o = ad \tan \frac{v}{w} \tag{1}$$

On performing experiments a difficulty arose in recognising large objects, as when they are up close they can appear smaller because the entire object does not fit in the image, giving a similar pixel area to a medium sized object. Therefore the robot tries to determine if the object seems to be extending outside of its vision. This is demonstrated in Figure 5 by the second interval, and is the number of sides. In this set of objects shown by the teacher, huge objects always extend out of the robots vision, and large objects sometimes do, but this is not the case for small and medium objects. We count the number of sides, as this might be used to determine very wide objects from generally huge objects.

[0 .. 33280][0 .. 1] → 0
    [2542 .. 4135][3 .. 4] → 1
    [481 .. 724][0 .. 2] → 2
    [159 .. 380][0 .. 0] → 3
    [36 .. 87][0 .. 0] → 4
0 = unknown, 1 = huge, 2 = large, 3 = medium, 4 = small

Figure 5: A learned RDR for classifying four sizes.

The categories are totally up to the teacher and are completely subjective. Over the course of the size experiments the ducks were named small, medium and large depending on their relation to the other objects being learned.

Shape

A main aim of the project is to make as few prior assumptions about the kinds of objects as possible. Therefore templates were not used for determining shape. Instead use simple calculations about the objects such as eccentricity. The gives us a basic idea if the object is short and long, tall and thin or fairly equal.

$$\varepsilon = \frac{\sqrt{x^2 - y^2}}{x} \tag{2}$$

In Figure 6, a round object, in this case a ball, has no eccentricity as the width is the same as the height. The tall and thin objects have quite similar eccentricity, and round and squat objects are also close. These two groups are very far apart however, so as you might intuitively think, a squat object is very unlikely to be classified as tall or thin, but it might be misclassified as being round. This similarity is likely because the duck, when front or back on, is quite round in shape.

[0 .. 100] → 0
    [79 .. 99] → 1
    [50 .. 67] → 2
    [1 .. 16] → 3
    [0 .. 0] → 4
0 = unknown, 1 = thin, 2 = tall, 3 = squat, 4 = round

Figure 6: A learned RDR for classifying four shapes.

Weight

The final feature was the objects weight, or 'pushability'. A humans concept of an object includes not only its physical characteristics, but also the kinds of things we can do with it. These kinds of object features are rarely included into object recognition systems. As we have a mobile robot, it is possible to exploit the extra information that it can get from interaction with objects.

It might be interesting to include a simple decision about whether the object can be pushed or not. However, the robot would only need to learn a mapping between two possible values of 'pushability', and two possible words. As this paper is about leaving the number of words open to the teacher, we instead look at the distance the object can be pushed.

To determine the object weight w, the robot first uses its infrared distance sensor to get the current distance d_1 to the object, it then uses its front paws to push the object. The robot then takes another distance measurement d_2 to find out how far the push caused it to move.

$$w = 10 \log(d_2 - d_1) \tag{3}$$

Due to a number of variables, the object will not always be moved the same distance with each robot push. Therefore, we use the *log* of the change in distance, so a medium change in distance is not considered entirely different to a large change.

[-16 .. 17] → 0
    [12 .. 16] → 1
    [-16 .. -3] → 2
    [10 .. 11] → 3
0 = unknown, 1 = light, 2 = heavy, 3 = knockover

Figure 7: A learned RDR for classifying three weights.

Figure 7 shows an RDR for three different kinds of objects. In this experiment, the light objects were different balls, which could be pushed a fair distance, the heavy objects were different solid rubber toys that could

Speaker	Speech and Action
Human	”This has the colour yellow.” H waves a yellow object at R.
Robot	”So, this is yellow.” R creates rule for yellow.
Human	”What colour is this?” H shows R another yellow object.
Robot	”Yellow?” R looks for any known pixels in image that match rules.
Human	”Yes, right” R computes the minimum cut using the known yellow pixels. R extends the rule for yellow using pixels found inside the area cut.
Human	”What colour is this?” H shows R an orange object.
Robot	”Yellow?” R has included orange pixels from the yellow object in its rules.
Human	”No, it is orange” R adds exception rules for orange.

Table 1: A typical interaction to teach colour names.

be moved very slightly, and the ’knockover’ objects were skittles, which could be pushed over.

2.6 Sample Interaction

A few words are known by the robot before we start learning new objects and features. The verbs “look” and “search” are inbuilt so that learning can be initiated. “Look” is used when the teacher wants to teach something directly and triggers the motion segmentation, and “search” is used to encourage the robot to go out and find its own examples. The searching capability is outside the scope of this paper. The attribute names are also know a priori so that we can specify what kind of feature detecting we should be using. These include “colour”, “size”, “shape”, and “weight”. Training occurs when the teacher actively names the object with sentences like “This is yellow in colour.”, and testing is achieved by presenting new objects and asking questions such as, “What colour is this?”.

Table 1 demonstrates a typical word-learning interaction between human and robot. The robot extends its rules for yellow too much and it contains orange as well. New rules are added to cope with this and the resulting rule set can be seen in Figure 2.

3 Results

Every trial began from scratch to demonstrate the system over the largest number of repeats. The features were re-taught with new examples from different objects, distances and viewing angles. The training examples were different objects from the test examples as much as possible. For example, the colour ’yellow’ was taught with a yellow duck, but tested with a yellow skittle. If not enough objects of that type were available, when it came to experiments with multiple tests per category, it was ensured that when re-seen they were at different distances and positions to demonstrate the greatest possible generalisation. Generally, the incorrect answers were false negatives. The robot said it did not know what the colour was, when in fact it had been taught that feature. There were few incidences of false positives, in which the robot labelled the feature incorrectly, and these occurred most often within the size feature.

Feature	Avg Accuracy	Tests
Colour	98.21% ±4.54	154
Size	98.75% ±10.4	82
Shape	96.21% ±15.6	120
Weight	88.69% ±18.4	47

Table 2: Total accuracy and number of object tests for the different features.

Colour words that were learned included pink, baby pink, red, blue, navy, grey, brown, green, yellow, orange, and blush. Size words that were taught included tiny, small, medium, big, large, and huge. Shape words that were taught included squat, fat, round, tall, thin, wide, flat, long. Weight words included light, easy, heavy, hard, feather, solid, pushable, and knockover. Clearly some of these words are synonyms, but the lists simply illustrate that the choice of word is not important.

The colour feature demonstrated the best accuracy. Although the average accuracy is slightly lower than that for the size feature, the standard deviation is substantially lower. The size and weight features had fewer tests because of the additional time required for the robot to gain the correct distance. They both require the robot to walk into a good position to get a infrared reading, centre the object in its vision and/or try and push the object with its paw. There were additional colour tests because we had objects of up to eight colours. The shape feature had the lowest accuracy and highest standard deviation. We hope to fix this by allowing for objects to have several shapes to compensate for different viewing angles. This is explained below in more detail.

Figure 8 compares the number of colour words in each experiment. For two and three colour words it was always 100% correct. The accuracy declined slightly as the number of words increased. It demonstrated a low of 93.8% when there were eight words, and was mainly due to the final two colours being a grey and a brown, which could be found in the background of the walls and

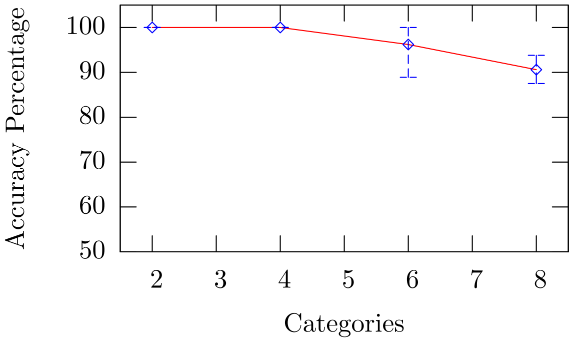

Figure 8: Colour Words.

floor. This caused smaller objects to be named as grey or brown when seen amount of large background.

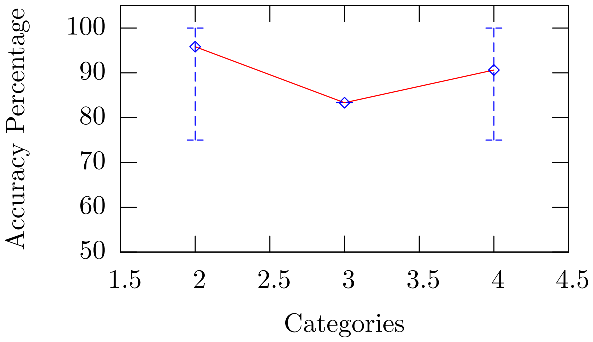

Figure 9: Size Words.

Increasing the number of words taught for the size feature had a less clear effect, Figure 9. At three words, there is a dip in naming accuracy, but this has a lower standard deviation that two or four words.

In Figure 10, for the shape feature we can see that as the number of training examples increased so did the naming accuracy. This similar effect was found for the size feature, but not for the colour feature. This occurs for the shape feature because the objects are not the same shape from all angles. As mentioned above the tests were always from different angles and distances to the training objects. As the robot is given more presentations of the object it can extend the rules to deal with this. The next stage of this project involves learning about entire objects. As we approach four training presentations we start to lose the benefit as we have most likely seen most of the very different shapes.

Performing these tests allowed us to discover that when teaching whole objects it will be desirable to allow for different shapes to match to the same object. For example, we could recognise a cylinder when it was round, or a wide rectangle, rather than trying to fit both shapes into one interval, which contains both and everything in between. The robot in these experiments, which are outside the scope of this paper, walks around the object collecting the different shapes and colours that can be associated with this object.

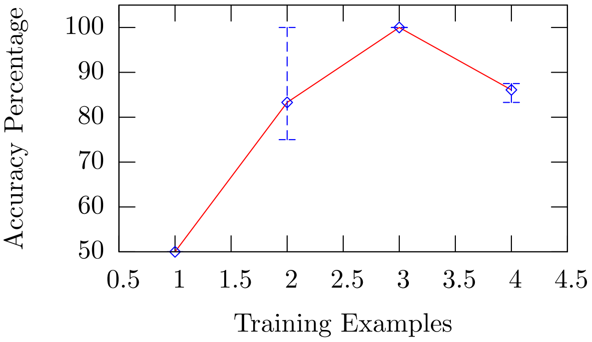

Figure 10: Shape Words.

Figure 11 shows a very similar curve for training examples for the weight feature. The weight feature benefits from several examples from the teacher as there are a range of distances the robot can push each object. The ball, for example, can be pushed quite far if the robot is in a good position when it hits it. However, if the robot is not lined up correctly, it may only go a short way.

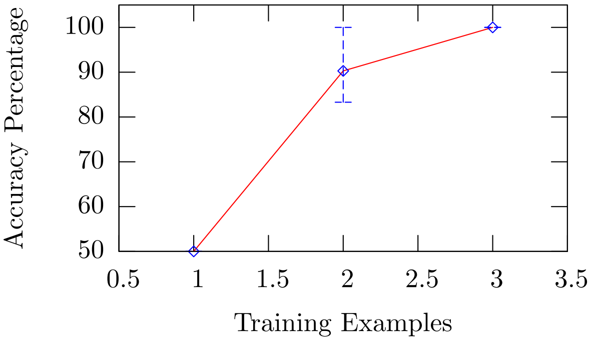

Figure 11: Weight Words.

4 Conclusions

Ripple Down Rules, motion segmentation and minimum/flow maximum cut work together well to extract features for word learning. They allow for as few assumptions about the objects to be made as possible and leave it open for the human teacher's interpretation of them. The Ripple Down Rules could be created successfully with very low involvement from the teacher, or expert. It was also shown that general features could be found in one object and applied to another object.

The next stage of this work involves learning about whole objects. After learning about some objects directly from the teacher, it seeks out new objects that may help to clarify its word meanings. In this final system, it is important that the human teacher is able to correct the robot when it names something incorrectly, and this is easily dealt with by Ripple Down Rules.

Using a mobile robot with limbs for manipulating objects allowed it to move towards finding affordances, as well as physical features of objects. A rich concept of objects that includes this kind of information will be necessary for controlling a robot through speech.

Acknowledgements

This work was supported by the Australian Research Council Centre for Excellence for Autonomous Systems.

References

[Boykov and Kolmogorov, 2001] An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 359-374, 2001.

[Compton and Jansen, 1988] P. Compton and R. Jansen. Knowledge in Context: a strategy for expert system maintenance *Second Australian Joint Artificial Intelligence Conference.* 292-306, 1988

[Fitzpatrick, 2003a] Paul Fitzpatrick. First Contact: an Active Vision Approach to Segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, 2003.

[Fitzpatrick, 2003b] Paul Fitzpatrick. From First Contact to Close Encounters: A developmentally deep perceptual system for a humanoid robot. In *PhD thesis*, Massachusetts Institute of Technology, 2003.

[Kerr and Compton, 2003] Julian Kerr and Paul Compton. Toward Generic Model-Based Object Recognition by Knowledge Acquisition and Machine Learning. In *Workshop on Mixed-Initiative Intelligent Systems, International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[Misra, Sowmya, and Compton, 2004] Avishkar Misra, Arcot Sowmya and Paul Compton. Incremental Learning of a Control Knowledge for Lung Boundary Extraction. In *Pacific Rim International Conference on Artificial Intelligence*, Auckland, 2004.

[Pelleg and Moore, 2000] Dan Pelleg and Andrew Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, 727-734, 2000.

[Scheffer, 1996] , Tobias Scheffer. Algebraic foundations and improved methods of induction or ripple- down rules. In *The Second Pacific Rim Knowledge Acquisition Workshop*, 279-292, 1996.

[Sammut, 2005] Claude Sammut. rUNSWift Robocup Report 2005. *to appear*, 2005

[Steels and Kaplan, 2002] Luc Steels and Frederic Kaplan. Bootstrapping Grounded Word Semantics. *Linguistic evolution through language acquisition: formal and computational models*, Cambridge University Press, 2002.