In: IEEE First International Conference on Cognitive Systems and Information Processing (CSIP 2012), AISC 215, pp. 881-819 Beijing, CH, December 15-17, 2012.

Object Learning with Natural Language in a Distributed Intelligent System – A Case Study of Human-Robot Interaction

Stefan Heinrich, Pascal Folleher, Peer Springstübe, Erik Strahl, Johannes Twiefel, Cornelius Weber, and Stefan Wermter

University of Hamburg, Department of Informatics, Knowledge Technology
Vogt-Kölln-Straße 30, D - 22527 Hamburg, Germany
{heinrich,6follehe,3springs,strahl}@informatik.uni-hamburg.de
{7twiefel,weber,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

Abstract. The development of humanoid robots for helping humans as well as for understanding the human cognitive system is of significant interest in science and technology. How to bridge the large gap between the needs of a natural human-robot interaction and the capabilities of recent humanoid platforms is an important but open question. In this paper we describe a system to teach a robot, based on a dialogue in natural language about its real environment in real time. For this, we integrate a fast object recognition method for the NAO humanoid robot and a hybrid ensemble learning mechanism. With a qualitative analysis we show the effectiveness of our system.

Keywords: Ensemble Learning, Human-Robot Interaction, Language

1 Introduction

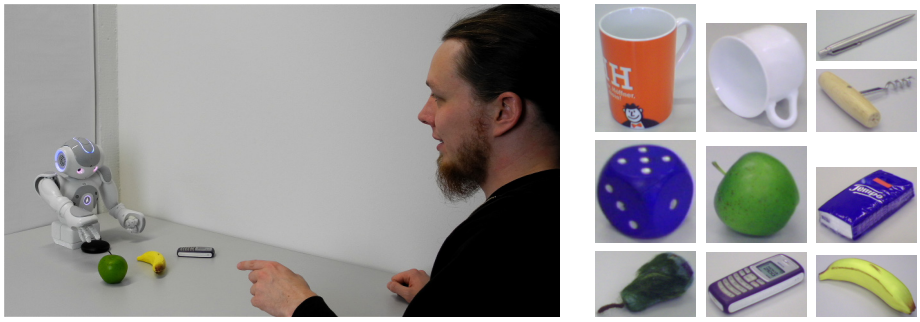
The interest in robots as assistants or companions has grown tremendously during the last years. Robots are developed to support humans in households as well as in healthcare and therapy [11]. In addition, research progresses in the direction of cognitive systems to understand cognitive functions in humans as well as to create robots that can interact with humans naturally [10].

For the development of an intelligent system that can fulfil these criteria, we have to bridge the large gap between the needs for human-robot interaction (for example based on a dialogue in natural language) and the technical capabilities of modern humanoid platforms and computing machines. The questions of how such a system can be designed and how state-of-the-art methods from machine learning and information processing can be integrated remains open [7].

To approach these questions, in this student's show case we developed a complex distributed system that is able to incorporate a humanoid robot, different standard machines and recent frameworks for various tasks. As a novel contribution we developed and included object detection and hybrid ensemble learning mechanisms that are able to operate in real time and within a real world environment. We show the effectiveness of these mechanisms in a qualitative analysis.

2 Scenario

Our research focuses on human-robot interaction in a real world scenario with real time conditions to learn about communication and grounding of language as well as about effective learned situated interaction [8]. Here a humanoid robot NAO[1] is supposed to learn cues about objects in its environment based on natural language and visual information, and to recognise and classify similar objects correctly (see Fig. 1a for an overview). The learning process is guided by a dialogue with a human teacher about some objects (compare Fig. 1b).



(a) Learner and teacher (b) Objects

Fig. 1. Scenario of learning objects by natural language in human-robot interaction.

The teacher can inform the robot about unknown objects, and is also able to confirm the correct classification and thus the correct pointing to objects, giving the robot the opportunity to become more certain with its decisions over time:

- **Teaching Dialog:** A user can request the robot to *learn*. The robot then asks what he is supposed to learn and the user states an object category (e.g. `<This is an apple>`). The robot then asks the user to verify the linguistic expression the robot has understood for the object in the field of view. After verification (e.g. `<Right>`) by the user the robot will learn the object.
- **Classification Dialog:** A user can also ask the robot to *classify* an object. The robot responds by reporting a description of the object in the field of view based on recently learned experiences.
- **Find Dialog:** In addition a user can request the robot to *find* an object among a number of different objects in the field of view. If the robot recognises the described object, then it will report a relative position and point to the object. Otherwise it will express his uncertainty about the requested object.

[1] The NAO is a 57 cm tall humanoid robot with, 25 *degrees of freedom* (DOF), two VGA cameras, and four microphones, developed for academic purposes – www.aldebaran-robotics.com

3 Architecture of the Distributed Intelligent System

The described scenario demands a lot of capabilities of an intelligent system: First of all the robot has to observe the scene and determine objects of various complexity under fairly different light conditions in real time. Secondly, the system has to provide reliable speech recognition and the ability to speak to the human in natural language. Thirdly, the system must learn objects very fast and also be scalable to a reasonable number of objects. Finally, the system has to incorporate all capabilities in a coherent interaction scheme.

To achieve the goals we set up a distributed system of up to 16 service nodes, written in ROS[2], both on the NAO robot and some standard PCs. The system can be divided into the four modules *core*, *vision*, *motion*, and *interface* (see Fig. 2 for an overview).

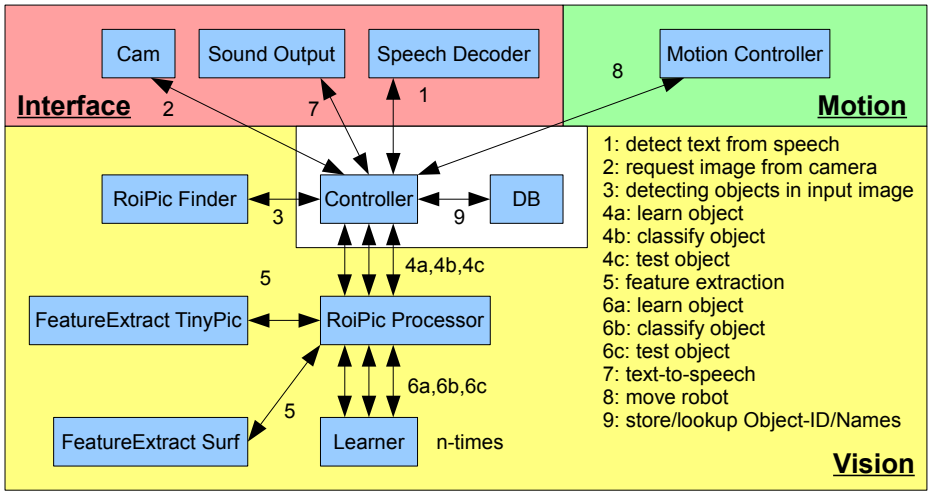


Fig. 2. An overview of the system components.

- **Core:** The heart of the system is the *Controller* that implements the logic of the dialogue by means of a finite state machine, set up by Smach[3]. A database is used to store training data of items that a user wanted the robot to learn and to allow for a qualitative evaluation.
- **Interface:** To interact with the real world the Controller can request the camera system to get real time images from the NAO, can request to recog-

[2] The *Robot Operating System* (ROS) is a middle-ware for the use with various robotic platforms; ROS supports different programming languages and multiple operating systems as well as multiple hardware elements – www.ros.org

[3] *Smach* is a python-based library for building hierarchical state machines – www.ros.org/wiki/smach/

nise speech with the employed PocketSphinx[4] with a hybrid decoder [5], and command the NAO's text-to-speech system to respond with speech.
- **Vision:** The Controller can request the vision system to determine a small extract from the image called *Region of Interest Picture* (RoiPic). For this task the *RoiPicFinder* isolates image regions by thresholding over certain ranges of hue, saturation, and value (HSV) for binarising in relevant parts and background, and applies the contour finder from OpenCV[5] for segmentation [9]. Note: Here, one also could use more precise but less fast methods like connected component labelling or clustering [4]. Finally, the RoiPicFinder computes the axis aligned bounding boxes, which are used to crop and return the RoiPic. Fig. 3a-c visualise the processing steps.
  Furthermore, the vision system offers a multiple-purpose feature extracting module that can determine various features from the RoiPic to return an input for a learner in the hybrid ensemble learning system. The learning system is realised by the *RoiPicProcessor* and can combine an arbitrary number of learners based on different features.
- **Motion:** The motion controller can be requested to move the robot body in the environment, e.g. to point to an object with the arm.

All modules are interconnected but distributed and autonomous, allowing to extend the system, e.g. with different feature extractors or to enrich the robot's behaviour by a more capable motion, thus offering a richer interaction with the environment. However, the central idea here is to research into learning objects by natural language, thus the focus of this study is on the hybrid ensemble.

4 Hybrid Ensemble Learning

For the learning we developed a hybrid ensemble system based on a set of neural associator networks called *Learners* [3]. Each network is a three-layer MLP that takes the result of a feature extractor as input and computes the classification *confidences* for a chosen number of classes as output. The ensemble votes for the class $c \in C$ with the highest confidence $o_{\text{ensemble},c}$, which is determined as follows:

$$o_{\text{ensemble},c} = \frac{\max_{l \in L} (o_{l,c} \cdot g_l)}{\sum_{l \in L} (o_{l,c} \cdot g_l)} \quad , \tag{1}$$

where for every Learner $l \in L$ the output o is weighted by a chosen value g.

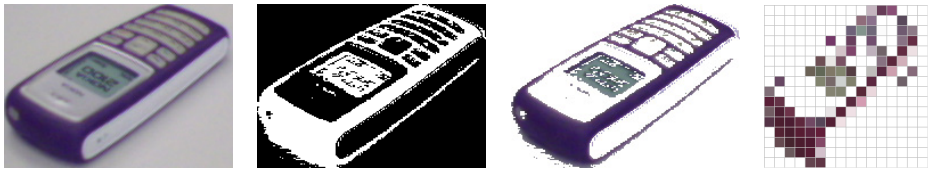
For the neural Learners we employed ENCOG[6], while for the feature extractors we developed three different types of features ourselves, as described below:

[4] *PocketSphinx* is an open source *automatic speech recognition* (ASR) system, optimised for hand-held devices and robots – www.cmusphinx.sourceforge.net

[5] The *Open Source Computer Vision* (OpenCV) library is a framework for state-of-the-art computer vision algorithms – www.opencv.willowgarage.com/wiki/

[6] *ENCOG* is a machine learning library focused on advanced neural networks and recent training methods – www.heatonresearch.com/encog/

Pixel Pattern Features: The simplest features we extract from a RoiPic is the pixel pattern of the object. To determine these features for the so called *TiniPic*, we scaled the isolated image from a RoiPic to the fixed size of 16×16 pixels with RGB values. The resulting 768 data points were normalised to floating point values within $[-1, 1]$ and could be fed into a Learner. Fig. 3 presents the steps to determine the scaled TinyPic.



(a) Region of interest (b) Threshold filter (b) Isolated object (c) Scaled

Fig. 3. Region of interest processed to determine a 16×16 pixel pattern.

Colour and Texture Features: For the human eye, colour and textural characteristics of an object are important [6]. Based on this bio-informed concept we developed a sophisticated extractor to determine twelve colour and texture features from a RoiPic as detailed in Tab. 1. For all features we normalised the values to the interval $[-1, 1]$ to be able to input them to a Learner.

Table 1. Developed Colour and Texture Features.

proportion of coloured pixels being of a certain colour (for six, nine, and twelve colours – each for 0° and 30°)	proportion of pixels that have colour information (exceeding saturation and brightness threshold)
sine of average colour of coloured pixels	*cosine* of average colour of coloured pixels
average brightness of object	average saturation of object
average brightness of coloured pixels	average saturation of coloured pixels
average grey value (average brightness of not coloured pixels)	proportion of pixels, which are part of the object (not background)
colour spectrum of coloured pixels	brightness spectrum of not coloured pixels

The "proportion of colour" features are calculated by dividing the colour space in six, nine, and twelve colours respectively. Further proportion of colour features are computed by shifting the HSV colour value by 30 before assigning it to the new colour space. The proportion of saturation and brightness is achieved by dividing their spaces in eight components. The test for matching the twelve colours is done by comparison to the HSV scale. The intervals for the colours are centred around 0° (red), 30° (orange), etc. and overlap as the intervals span ±20° from the centre. With this method a pixel can be both e.g. red and orange, which is close to what happens in human perception.

Standardised SURF features: The conventional *SURF* [2] algorithm is able to robustly detect and recognise local features of an object in an image. However, the format of the conventional SURF features makes it impossible to combine SURF with many other learning methods, e.g. associator networks, because the dimensionality of the representation for a specific object is not known a-priori. Usually SURF results in a very large set of features for a complex object and a very small set of features for plain objects. To overcome this issue we standardised the output of the SURF extractor as follows:

We reduced the 64 double values to eight double values by summing up blocks of eight numbers and determined a seven bit number, where each bit represents a rise (bit set to 1) or a fall (bit set to 0). The remaining highest bit in the byte was determined by the Laplacian, which was calculated by the SURF extractor. The resulting kind-of "hash" (256 bits) is consistent in sparseness, leads to an unique characterisations of an object, and can be fed to a Learner.

5 Evaluation

To evaluate the system we tested its behaviour in a number of dialogues with different human teachers and observed a very natural interaction and good learning success without notable delays: The computations of the system are performed in parallel to the speech output, providing a real-time response at any time.

To offer a more comparable evaluation we also ran several experiments to quantify the object detection and object learning capabilities. For all experiments we set up the system with an ensemble consisting of five colour and texture Learners, three pixel pattern Learners, and two standardised SURF Learners. The neural networks underlying these *classifiers* have been specified with 100 hidden nodes, 21 output nodes, sigmoidal transfer functions, and randomised weights in $[-0.25, 0.25]$. They have been trained with RPROP [3] for either a maximum of 100 epochs or until a mean error of at most $\epsilon = 0.01$ was reached.

5.1 Object Detection

To evaluate the quality of our detected objects by means of the determined region of interest (dimensions and position), we developed the following metric:

$$q = \frac{A_{RF} - |A_{RF} - A_{GT}|}{A_{GT} + |A_{RF} - A_{GT}| + d_e(POS_{GT}, POS_{RF})} \quad , \tag{2}$$

where A is the area in pixels, d_e the euclidean distance, POS the bounding box reference point, RF the results of the RoiPicFinder, and GT the ideal result.

For all ten objects we collected 20 samples covering different rotation and scaling as well as different lighting conditions in our standard lab environment (compare Fig. 1) and ran two experiments. In the first experiment we employed the near optimal grey scale value during the thresholding step, while in the second experiment we used HSV values. For five representative objects the results of the quality of the obtained regions of interest are shown in Tab. 2, pointing out that our developed method led to

a) a good object detection for most objects – except for objects with high diversity in the texture – with near optimal values (in 0.074 seconds), and
b) an overall very good object detection with HSV values (in 0.71 seconds).

Table 2. Results for the quality q of determined regions of interest. For thesholding different values have been used: near optimal grey scale (left) and HSV (right).

Object Class	average	min	max
Apple	0.886	0.552	0.989
Banana	0.111	-0.383	0.986
Dice	0.903	0.683	0.999
Mobile	0.793	0.446	0.998
Pear	0.690	0.252	0.996

Object Class	average	min	max
Apple	0.945	0.675	0.997
Banana	0.859	0.671	0.994
Dice	0.960	0.909	0.987
Mobile	0.949	0.844	0.996
Pear	0.959	0.824	1.000

5.2 Object Learning and Generalisation

For testing the generalisation capabilities we used the standard metrics precision $p_{\text{precision}} = tp/(tp + fp)$ and recall $p_{\text{recall}} = tp/(tp + fn)$, where we defined all correct classifications as tp (true positives), all classifications for an incorrect class as fp (false positives), and all classifications with a *confidence* $o < 0.45$ as fn (false negatives). For every object we divided the set of samples in a training set with 15 samples and a test set with 5 samples and conducted two experiments. In the first one we trained and tested with all objects, while in the second we trained and tested only with the three very similar objects "Dice", "Mobile", and "Tempo". The results show that for a diverse set of objects the colour and texture classifiers achieve very high results, thus performing still high for similar objects (see Tab. 3). The hybrid ensemble leads to high up to very high results in all settings.

Table 3. Classification results on the test set for all (left) and similar (right) objects.

Classifier	$p_{\text{precision}}$	p_{recall}
Pixel Pattern	0.590	0.976
Color & Texture	0.984	1.000
Standardised SURF	0.391	0.895
Ensemble	0.979	0.939

Classifier	$p_{\text{precision}}$	p_{recall}
Pixel Pattern	0.644	1.000
Color & Texture	0.893	1.000
Standardised SURF	0.621	1.000
Ensemble	1.000	1.000

6 Conclusion

In this paper we investigated the needs for human-robot interaction and developed a distributed intelligent system to enable a humanoid robot to learn about its environment by a human teacher via a dialogue. The combination of recent frameworks and a number of specially developed methods for object detection and learning led to a system working in real time and in a real environment.

For the object detection we learned that using simple well elaborated methods already can alleviate the problem of real time processing tremendously. Finding good parameters e.g. for thresholding still is an issue but can be overcome by more recent methods once they can be computed very fast [4]. The learning with hybrid ensembles works well and taught us to take very diverse classifiers into account, which are also inspired by human capabilities, e.g. the processing of texture information [6]. A very diverse or even multi-modal set of classifiers needs to be integrated in a smart way, but this can be solved with other learning mechanisms on top, e.g. advanced self-organising networks [1].

In the future we aim to push further the natural interaction of the robot. A robot could, for instance, explore a whole room on its own and learn about objects by touching and manipulating them. This can help to understand the behaviour of young children and the need for autonomous learning systems [10].

Acknowledgments

The authors would like to thank Sven Magg and Nils Meins for very inspiring as well as very helpful discussions. This work has been partially supported by the KSERA project funded by the European Commission under n° 2010-248085 and by the RobotDoC project funded by Marie Curie ITN under 235065.

References

1. Bauer, J., Weber, C., Wermter, S.: A som-based model for multi-sensory integration in the superior colliculus. In: Proc. 2012 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE, Brisbane, AUS (Jun. 2012)
2. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. Computer Vision and Image Understanding 110(3), 404–417 (2006)
3. Du, K.L., Swamy, M.N.S.: Neural Networks in a Softcomputing Framework. Springer New York (2006)
4. He, L., Chao, Y., Suzuki, K., Wu, K.: Fast connected-component labeling. Pattern Recognition 42(9), 1977–1987 (2009)
5. Heinrich, S., Wermter, S.: Towards robust speech recognition for human-robot interaction. In: Proc. of the IROS2011 Workshop on Cognitive Neuroscience Robotics (CNR). pp. 29–34. San Francisco, CA, USA (Sep. 2011)
6. Mel, B.W.: SEEMORE: Combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. Neural Computation 9(4), 777–804 (1997)
7. Pfeifer, R., Bongard, J., Berry, D.: Designing intelligence: Why brains aren't enough. GRIN Verlag (2011)
8. Spranger, M., Loetzsch, M., Steels, L.: A perceptual system for language game experiments. In: Language Grounding in Robots, pp. 89–110. Springer, NY (2012)
9. Suzuki, S., Abe, K.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, & Image Process. 30(1), 32–46 (1985)
10. Vernon, D., von Hofsten, C., Fadiga, L.: A Roadmap for Cognitive Development in Humanoid Robots. Springer-Verlag Berlin Heidelberg (2011)
11. Wada, K., Shibata, T.: Social and physiological influences of living with seal robots in an elderly care house for two months. Gerontechnology 7(2), 235 (2008)