FORMATION OF ADJECTIVE, NOUN AND VERB CONCEPTS THROUGH AFFORDANCES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR YÜRÜTEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ONUR YÜRÜTEN

Signature :

ABSTRACT

FORMATION OF ADJECTIVE, NOUN AND VERB CONCEPTS THROUGH AFFORDANCES

Yürüten, Onur
M.S., Department of Computer Engineering
Supervisor : Asst. Prof. Dr. Sinan Kalkan
Co-Supervisor : Assoc. Prof. Dr. Erol Şahin

June 2012, 75 pages

In this thesis, we study the development of linguistic concepts (corresponding to a subset of nouns, verbs and adjectives) on a humanoid robot. To accomplish this goal, we use affordances, a notion first proposed by J.J. Gibson to describe the action possibilities offered to an agent by the environment. Using the affordances formalization framework of Sahin et al., we have implemented a learning system on a humanoid robot and obtained the required data from the sensorimotor experiences of the robot. The system we developed (1) can learn verb, adjective and noun concepts, (2) represent them in terms of strings of prototypes and dependencies based on affordances, (3) can accurately recognize the concept of novel objects and events, and (4) can be used for tasks such as goal emulation and multi step planning.

Keywords: Concept Learning, Generalization, Humanoid Robot, Goal Emulation

ÖZ

SIFAT, NESNE VE FİİL KAVRAMLARININ SAĞLARLIK YOLUYLA OLUŞUMU

Yürüten, Onur
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Sinan Kalkan
Ortak Tez Yöneticisi : Doç. Dr. Erol Şahin

Haziran 2012, 75 sayfa

Bu tezde insansı bir robot üzerinde dil kavramları (nesne, fiil ve sıfat) geliştirme problemi üzerinde çalışılmıştır. Bu amaca ulaşmak için J.J. Gibson tarafından ortamın ajana sunduğu hareket olasılıklarını tanımlamak için ortaya atılan sağlarlık fikri üzerinden hareket edilmiştir. Sahin et.al.'ın sağlarlık için ortaya koyduğu formalizasyon çerçevesinde, bir insansı robot üzerinde bir öğrenme sistemi geliştirilmiş ve gerekli veriler bu robotun duyu-motor deneyimleri üzerinden toplanmıştır. Geliştirilen sistem (1) Fiil, sıfat ve nesne kavramlarını öğrenebilmekte, (2) Bu kavramları, sağlarlıklara dayalı prototip ve bağıntı dizilimleri şeklinde ifade edebilmekte, (3) görülmemiş veriler sunulduğunda güçlü ve isabetli, ve (4) hedef öykünme gibi uygulamalarda kullanılabilmektedir.

Anahtar Kelimeler: Kavram Ögrenme, Genelleme, İnsansı Robot, Hedef Emülasyonu

For All Beloved

ACKNOWLEDGMENTS

I would like to acknowledge the support of TÜBİTAK (The Scientific and Technological Research Council of Turkey) BİDEB 2210 graduate student fellowship and this thesis is supported by TÜBİTAK under the project with id 109E033 and European Commission under the ROSSI project with id FP7-216125.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES xi

LIST OF FIGURES xiii

CHAPTERS

1 INTRODUCTION 1

1.1 Problem Definition 1

1.2 Proposed Approach 2

1.3 Contributions 3

1.4 Outline of the Thesis 5

2 BACKGROUND AND LITERATURE SURVEY 7

2.1 Developmental and Cognitive Robotics 7

2.2 Concepts 10

2.2.1 Theories of Concept 10

2.2.2 Symbol Grounding, Concepts and Language 12

2.3 Affordances 14

2.4 Curse of Dimensionality and Feature Selection 16

2.5 Support Vector Machine for Supervised Learning 18

2.6 Self Organizing Maps 19

3 SETUP AND METHODS 21

3.1 Setup 21

3.1.1 The iCub Humanoid Platform 21

3.1.2 Behaviors 21
3.1.3 Yet Another Robotic Platform (YARP), Robotic Operating System (ROS) and iCub Software 22
3.1.4 Perception 23
3.1.4.1 Kinect 23
3.1.4.2 VisualEyez VZ4000 23
3.1.4.3 Features for Objects and Effects 24
3.2 Data Collection 26
3.3 Affordance Learning 27
3.4 Category Learning 27
3.4.1 Verb Learning Model 27
3.4.2 Adjective Learning Model 30
3.4.3 Noun Learning Model 32
3.4.4 Conceptualization 33
3.4.4.1 Effect Prototype Extraction 33
3.4.4.2 Adjective Prototypes 34
3.5 Metrics and Evaluations 34
3.6 Multi-Step Planning from Verb Concepts 36
3.7 Methodology Summary 39
4 EXPERIMENTS AND RESULTS 41
4.1 Affordance Learning Results 41
4.2 Verb Concepts 42
4.2.1 Learning Verb Categories 42
4.2.2 Representing Verb Categories 42
4.2.3 Interpreting Novel Interactions 43
4.3 Adjective Concepts 43
4.3.1 Learning Adjective Categories 43
4.3.2 Representing Adjective Categories 45
4.3.3 Adjectives of Novel Entities 48
4.4 Noun Concepts 48
4.4.1 Learning Noun Categories 48

4.4.2 Representing Noun Categories 50
4.4.3 Nouns of Novel Entities 50
4.5 Multi-Step Planning 50
4.6 Computational Complexity 56
5 DISCUSSION AND CONCLUSION 57
5.1 Future Work 59
REFERENCES 61
APPENDICES
A USED LEARNING ALGORITHMS 71
A.1 Relief and ReliefF Algorithms 71
A.2 Self Organizing Maps - SOM 71
A.3 Robust Growing Neural Gas - RGNG 72
B SOFTWARE PLATFORMS 74
B.1 YARP - Yet Another Robotic Platform 74
B.2 ROS 74
B.3 PCL - Point Cloud Library 75

LIST OF TABLES

TABLES

Table 4.1 Average, maximum and minimum prediction results for each behavior with respect to 5-fold cross validation. 41

Table 4.2 The effect prototypes. The effects are abbreviated as explained in Section 4.2.2 44

Table 4.3 Distances computed for effect instances in Figure 4.1 using the Equation 3.11 (first rows, abbreviated as d_{mm}), Equation 3.12 (second rows, abbreviated as d_e) and Equation 3.13 (third rows, abbreviated as d_{pm}). The label of the cluster with the smallest distance value is given as the prediction to the novel effect. The correct predictions are marked with bold green, whereas false predictions are marked with italic red. The effects are abbreviated as explained in Section 4.2.2 . 45

Table 4.4 Average prediction results for the three adjective models in Sect. 3.4. . . . 45

Table 4.5 The prototypes of adjectives for the model A_{48}-AL ($\mathcal{M}_a^1$). TG: Top Grasp, SG: Side Grasp, PR: Push Right, PL: Push Left, PF: Push Forward, PB: Pull. For each behavior, there are eight effect categories: *a*: Moved Right, *b*: Moved Left, *c*: Moved Forward, *d*: Pulled, *e*: Knocked, *f*: No Change *g*: Grasped, *h*: Disappeared. 47

Table 4.6 The prototypes of adjectives for the model A_8-AL ($\mathcal{M}_a^2$). The effects are abbreviated as explained in Section 4.2.2 48

Table 4.7 Predicted adjectives for novel objects using 3 different models (bold labels denote correct classifications). 49

Table 4.8 The noun strings. The labels 0, + and * correspond to *Consistently Small*, *Consistently Large* and *Inconsistent* features. 51

Table 4.9 Noun prediction for novel objects using 3 different models. Bold labels denote the correct classifications with respect to the ground truth (see Table 4.7 for pictures of the objects). 52

LIST OF FIGURES

FIGURES

Figure 1.1 **(a)** Different behaviors through which one can push an object. **(b)** Cups with a high variance of perceptual features. 2

Figure 2.1 The HiTEC model. Figure taken from [1]. 9

Figure 2.2 The ABL model of [2], exemplifying how a verb (*Drink* in this context) can be understood. 10

Figure 2.3 The depiction of the affordance formalization framework. The verb concepts can be obtained via generalizations of affordance relations over the effects (as in Figure 2.3(b) and the noun and adjectives concepts can be obtained via generalizations over entities (as in Figure 2.3(c)).. 15

Figure 2.4 The hierarchical clustering described by Ugur et al. [3], which groups the information derived from interacting with the environment into effect categories. In this study, the effect categories that have small sizes are discarded automatically. 17

Figure 2.5 Using kernel functions, the space can be transformed such that the mapped data is linearly separable. Then, the optimally separating hyperplane can be calculated. 18

Figure 2.6 The learning architecture proposed by Morse et al. [4]. The grids of grey balls conceptually represent self-organizing maps and each grey ball is a node (unit). 20

Figure 3.1 The iCub Humanoid platform with a Kinect sensor at side. 22

Figure 3.2 The Kinect sensor. 23

Figure 3.3 The VisualEyez VZ4000 tracking device. 24

Figure 3.4 Snapshots from *rviz* ROS visualization tool. The perception system efficiently detects and displays the objects on the table. iCub is represented by primitive shapes which becomes useful if the 3D points corresponding to the robot body is to be filtered out. 25

Figure 3.5 Overview of the system. iCub perceives the environment and learns the affordances. From the affordances learned from the perceptual data, adjective classifiers are learned. 26

Figure 3.6 The objects in our dataset. 26

Figure 3.7 The mechanism for affordance learning. We collect instances of effects and give them effect labels. These labels are in turn used to train SVMs, and separately, forming effect clusters. 28

Figure 3.8 The mechanism with explicit behavior information for adjective and noun learning. Each adjective and noun SVM classifier is fed with 48 dimensional affordance vectors, which is a concatenation of 8-dimensional affordance estimate vectors obtained from each behaviorwise SVM. 31

Figure 3.9 The mechanism with no explicit behavior information for adjective and noun learning. Each adjective and noun SVM classifier is fed with 8 dimensional affordance vectors, where each dimension corresponds to the highest estimate obtainable from the given behaviorwise SVM. For example, If the highest estimate for a given object from Push-left SVM is 0.95 (of *moved left* effect), then the corresponding dimension in the affordance vector would be set to 0.95. 32

Figure 3.10 The simple noun and adjective learner. Perceptual information are directly fed to the SVM classifiers. 36

Figure 3.11 A sketch of how the future states are calculated. First, the current state of the object is fed to the trained SVM for the each behaviour. Then, the predicted effect's prototype is determined by applying the Equation 3.11 for each effect prototype. The mean value of the found effect is added on the initial features, with the exception of the inconsistent features, and the predicted future state is found. After this application, the predicted future state can be compared with other states; but the inconsistent features of the applied effect (denoted as black columns in predicted after-state) is excluded from the comparison calculations. 37

Figure 3.12 A simple depiction of how the planning is performed using the combinations of effects in the repertoire. At each step, the prediction block described in Figure 3.11 is applied for each behaviour. Once a future state close enough to goal state is obtained, the search is terminated. 39

Figure 4.1 Some novel instances for effects. The effect is simply the difference of final and initial states of the given object. The novel effect predictions with different distance metrics for these instances are conveyed on Table 4.3 46

Figure 4.2 After learning nouns and adjectives, iCub can refer to an object with its higher level representations or understand what is meant if such representations are used by a human. 47

Figure 4.3 A sample execution of a multi step planning. First, the start position (4.3(a)) and final position is shown (4.3(b)). The robot plans out and applies push right (4.3(c)) and push forward (4.3(d) When a simple Euclidian distance is used, such a sequence of behaviors could not be planned). 52

Figure 4.4 A multi-step planning trial with the modified Mahalanobis distance metric. The behaviors are abbreviated as *PR* (*push-right*), *PL* (*push-left*), *PF* (*push-forward*), *PB* (*pull*), *TG* (*top-grasp*), *SG* (*side-grasp*). The planner successfully terminates with a reasonably small sequence of behaviors executed. The trial is also visualized in Figure 4.3 53

Figure 4.5 Multi-Step planning trial with pure mahalanobis distance. The behaviors are abbreviated as *PR* (*push-right*), *PL* (*push-left*), *PF* (*push-forward*), *PB* (*pull*), *TG* (*top-grasp*), *SG* (*side-grasp*). The initial and the target states for the objects are the same with the one provided in planning in Figure 4.3. Since the distance calculations yield wrong results, the search does not terminate with success. . . . 54

Figure 4.6 Multi-Step planning trial with pure euclidean distance. The behaviors are abbreviated as *PR* (*push-right*), *PL* (*push-left*), *PF* (*push-forward*), *PB* (*pull*), *TG* (*top-grasp*), *SG* (*side-grasp*). The initial and the target states for the objects are the same with the one provided in planning in Figure 4.3. Since the distance calculations yield wrong results, the search does not terminate with success. . . . 55

Figure 5.1 The proposed context-aware adjective learning system. iCub learns adjectives based on the affordance predictions and contextual features. 60

CHAPTER 1

INTRODUCTION

In this section, we describe the problem of interest, and state the contributions provided in this thesis.

1.1 Problem Definition

In near future, robots are expected to be part of our daily life, requiring seamless communication with humans using natural language. This requires the robot to comprehend and utilize language structures such as nouns, verbs, adjectives, adverbs, etc. In order to tackle this ambitious challenge, robots should have the capability to perceive its environment, to generalize from their sensorimotor interaction and also to communicate about what they perceive and cognize. To have human-like perceptual and cognitive abilities, a robot should be able (i) to relate its symbols or symbolic representations to its internal and external sensorimotor data/experiences, i.e., it should be able to address *the symbol grounding problem* [5], and (ii) to conceptualize over raw sensorimotor experiences towards abstract, compact and general representations. In this thesis, we address these two issues.

The current study was carried out within the ROSSI project [6], one objective of which is the development of concepts mediated by verbs and nouns in a robot to enable simple forms of communication with humans. For example, in order to carry out a simple command such as "push the cup", the robot needs to understand what it means to "push" an object, as well as what a "cup" is. As shown in Figure 1.1(a), there are several ways (behaviors) for pushing a cup, for which reason it is not suitable to link a verb with a single behavior. Furthermore, as exemplified in Figure 1.1(b), one can name many objects as "cup", whereas it is difficult to

find out a common pattern solely from their appearances. On the other hand, cups do share some common properties; for instance, all cups can be filled with some liquid. This suggests that we, as humans, conceptualize the world around us by utilizing the functional information we infer from our experiences in addition to appearances. Thus, a robot should be able to tap such critical information from its environment in order to acquire an understanding of natural language used by humans.

(a) (b)

Figure 1.1: **(a)** Different behaviors through which one can push an object. **(b)** Cups with a high variance of perceptual features.

1.2 Proposed Approach

We use iCub, a child-like humanoid platform for demonstrating the emergence of some noun, adjective and verb concepts. Towards this end, iCub interacts with different objects in its environment with a fixed set of behaviors. From these interactions, it forms categories[1] over the behaviors (corresponding to verb concepts) and over the objects (based on what they afford and how they look, corresponding to noun and adjective concepts). With a prototype-based representation proposed by Atil et al. [7] and Dag et al. [8], we construct representations for these categories over behaviors and objects, which basically correspond to verb, noun and adjective concepts.

[1] We define *concept* as the representation that describes a category

Our approach towards verbs is to link a verb with an effect[2] rather than a single behavior (as shown in Figure 1.1(a)). For this reason, we generalize over behaviors based on their effects and call these generalizations "verb concepts". For example, we propose that the robot can represent the verb "to push" by a string like (in a simplified setting where the perceptual state of the environment is represented by the 3D position (xyz) and the orientation (θ) of the object as $xyz\theta$):

$\mathbb{0}\mathbb{0}+\mathbb{0}$, which means an increase (+) in the z-coordinate of the object's position and no change ($\mathbb{0}$) in other feature elements, which represent the other properties of the object. There may be several behaviors for the robot for achieving the effect $\mathbb{0}\mathbb{0}+\mathbb{0}\mathbb{0}\mathbb{0}\mathbb{0}\mathbb{0}$ and we suggest that all of them can correspond to the verb "to push".

As for adjectives and nouns, in this thesis, we address only a subset of them. The subset of adjectives that we address in this thesis is based on what objects afford. For example, a ball affords rolling whereas a box does not afford it, and with the set of affordances an object affords and does not afford, we form categories over objects and construct representations from these categories to form adjective concepts. On the other hand, the subset of nouns that we address in this thesis is based on the appearance of the objects: by directly taking the perceptual features from the object, we form categories over the objects and construct representations from these categories to form noun concepts.

Based on the views of J. J. Gibson (who see perception as the process of discovering what the environment affords [9]) and E. J. Gibson (who see affordances as an important tool in finding out the invariance and invariant representations of the environment [10]), we base our approach on affordances and for this end, we make use of the affordance formalization framework of Sahin et al. [11], which was also employed in the works of Atil et al.[7] and Dag et al.[8].

1.3 Contributions

Corresponding to the issues mentioned in Section 1.1, this thesis presents the following contributions:

[2] We define *effect* of a behavior as what the behavior changes in the environment

- The works of Dağ [8] and Atıl [7] had shown that *verb* concepts and *object* categories can be derived by utilizing affordances in different computational models. Extending these, we present a system that not only can learn *noun* and *adjective* categories, which correspond to the object categories of Dag [8] and Atıl [7], but can also conceptualize them over prototypical representations (just like verbs).

- In forming object categories, Dağ and Atıl used a vector for each object that included the enumerations of effects obtained by applying each behavior to the object. We instead provide a probability estimate for each possible effect for each applied behavior. With these estimates, we are able to access the explicit behavior information and enhance the representation power of affordances. Our observations show that this approach allows an efficient way to capture many-to-one and one-to-many mappings of entity-behavior-effect relations. With this, accurate representations of adjectives and nouns can be obtained.

- We claim that some adjectives include functional information about the entities in the environment. We verify our claim by showing that affordance-based adjective learning outperforms simple appearance-based adjective learning in terms of both average prediction and novel adjective predictions.

- We show that verb concepts can be efficiently utilized in multi-step planning. Specifically, by obtaining the string representations that describe the changes induced by behaviors in the perceptual vectors describing the environment, we can discard the inconsistent features in distance calculations for more accurate and computationally cheaper planning. We demonstrate the superiority of our approach by comparing it with non-prototype representations. Our findings function as evidences for the advantage of Prototype-Based view of concepts over Exemplar-Based view of concepts (See Section 2.2.1).

The work presented in this thesis has appeared in the following publications:

- Onur Yürüten, Kadir F. Uyanık, Yiğit Çalışkan, Asil Kaan Bozcuoğlu, Erol Şahin and Sinan Kalkan, Learning Adjectives and Nouns from Affordances on the iCub Humanoid Robot. 12th International Conference on Adaptive Behaviour (2012).

- Sinan Kalkan, Nilgün Dag, Onur Yürüten, Anna M. Borghi and Erol Şahin, Verb Concepts from Affordances. Journal of Interaction Studies, Under Revision.

- Kadir F. Uyanık, Yiğit Çalışkan, Asil Kaan Bozcuoğlu, Onur Yürüten, Sinan Kalkan and Erol Şahin. Learning Social Affordances and Using Them for Planning. IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, 2012, Submitted.

- Kadir F. Uyanık, Onur Yürüten, Sinan Kalkan ad Erol Şahin. Towards Context-aware Adjective Learning. ICRA 2012 Workshop: The Future of HRI - Paving the way to next generation of HRI.

1.4 Outline of the Thesis

In Chapter 2 (Background and Literature Survey), we present the literature that surrounds our work of interest. We firstly mention literature that cover language concepts and sensorimotor experiences in fields of philosophy, neuroscience, and developmental robotics. Then, we shift towards more specialized literature based on affordances. Specifically, we focus on the notion of affordances, its different versions of formalization and applications. We stress the works that link affordances with concepts. We finally investigate on the learning techniques that are relevant to our approach.

In Chapter 3 (Methods), we describe the software (network structure, software architecture, feature extraction methods) and hardware (the iCub humanoid platform, VisualEyez VZ4000 and Kinect) used in our setup. Then, we lay out our strategy in forming noun, adjective and verb concepts and explain the rationale in linking them with affordances.

In chapter 4 (Experiments and Results), we show the learned concepts. We rigorously analyze our strategy with general performance differences against basic learning schemes, testing how robust the system reacts to novel data, and measuring prediction accuracies varying number of features. As a final remark, we show that the composition of actions through learned concepts yield a useful planning scheme in goal emulation.

Finally, in chapter 5 (Discussions and Conclusions), we discuss on the outcomes of our experiments for an overall evaluation of our work and point out the possible directions in which

this thesis can be improved.

CHAPTER 2

BACKGROUND AND LITERATURE SURVEY

In this chapter, we present the relevant literature.

2.1 Developmental and Cognitive Robotics

Developmental robotics is a new branch in robotics that inspire from the developmental psychology. The field assumes that the cognitive structures emerge in the wake of physical and social interactions [12]. There are some standard principles and methodologies being established. For example, Stoytchev mentions five principles, namely, **verification, embodiment, subjectivity, grounding** and **incremental development** [13]. These principles can be summarized as follows:

- *Verification*: This principle lies at the heart of developmental robotics, binding all other principles. It states that an Artificial Intelligence (AI) system must only work on propositions and knowledge that it can itself produce and verify.
- *Embodiment*: This principle simply states that to be able to verify any obtained knowledge, a cognitive agent must be able to do physical action. Hence, the agent must be embodied. As such, the robots obviously are good candidates to be cognitive agents. This principle originates from the ideas that reject the standing of Good-Old fashioned AI, where the systems would be disembodied and existed as pure codes.
- *Grounding*: This principle states that there must exist some axiomatic rules for the verification procedure. Thus, with grounding, what constitutes a successful verification can be determined. As discussed in Section 2.2.2, many agree that the grounding is

achieved with sensorimotor experiences.

- *Subjectivity*: With close connections to sensorimotor experiences, the principle of subjectivity states that the understanding and representation of the world is subject to the cognitive agent's history of experience. Since, along with some external events, the experiences are governed and limited by the sensor and motor capabilities, it is only natural to see that this representation is unique to the cognitive agent.

- *Incremental Development*: This principle states that the level of complexity for perception, action and understanding can not emerge all at once, but develop step by step. Stoytchev [13] exemplifies this by remarking that one can only learn to walk after learning how to crawl, and learn to read after learning to recognize individual letters.

Some studies demonstrate these in some practical applications such as pressing the doorbells [14] and odd-one-out game [15]. In the study of Sukhoy et.al. [14], a robot is programmed with three different exploration strategies (random, stimulus-driven and uncertainity driven) to learn to press on buttons. The second and third strategies involved incremental utilization of previous experiences and, intriguingly, performed much better than random exploration. In the study of Sinapov et al. [15], the objects are categorized with respect to the responses acquired by interacting a robot with them (the same one with [14]). These two studies present an important exemplification of the importance in acquiring functional properties through sensorimotor experiences.

Apart from these examples, the last decade has seen many approaches for cognitive and developmental learning architecture [1, 16], which deal with learning high level abstractions. Some of them even go further and demonstrate their study on well-known psychological phenomena. For example, Haazebroek et al. [1] propose an architecture called HiTEC that utilizes the inferences from cognitive psychology to simulate the Simon Effect[1]. This architecture is composed of levels of Task, Feature and Sensory-Motor that provide feedback to each other (Figure 2.1). These levels are composed of codes that are designed to provide feedback to either excite or inhibit activities of interconnected levels. This architecture strongly resembles to a neural network system. Incidentally, they also provide the implementation in terms of

[1] The Simon Effect is a psychological phenomenon first reported by Simon and Wolf [17]. According to their study, the location of the stimulus affects the reaction times. Although this location might be irrelevant in some tasks, this suggests that the neural architecture in our brain links symbols (stimulus) with sensorimotor experiences that produce responses

artificial neural networks.

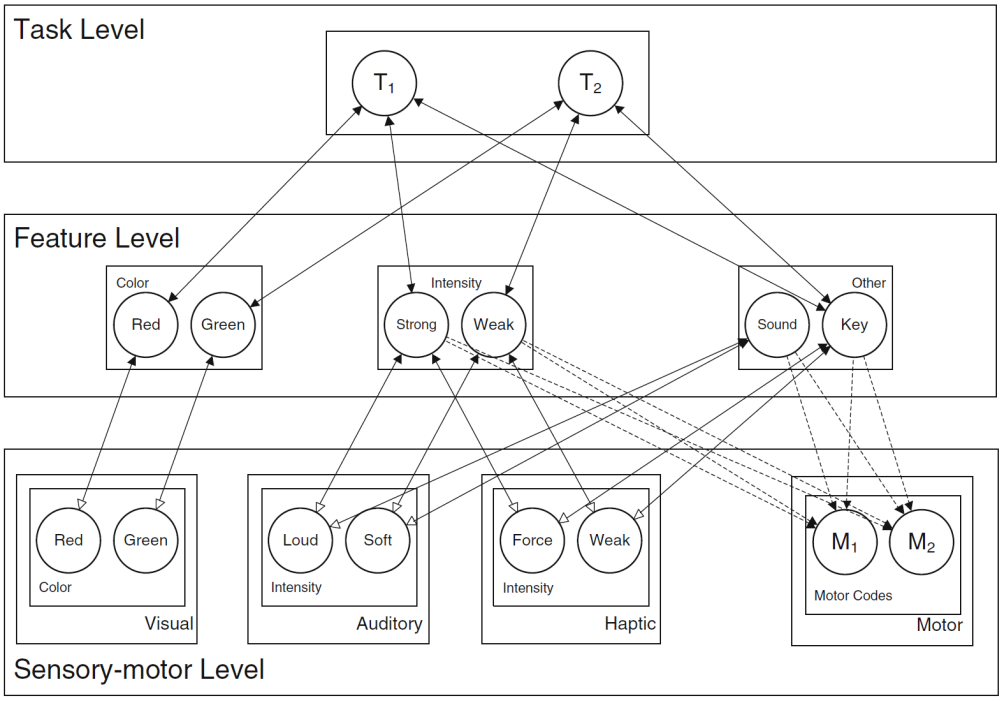


Figure 2.1: The HiTEC model. Figure taken from [1].

Some of these studies are closely related with our focus in this thesis, i.e., language grounding. Glenberg and Gallese [2] propose a theory that emphasizes exploiting goal-oriented actions for language learning. In this theory, Glenberg and Gallese account for language acquisition (specifically verbs and nouns), comprehension and production. They do so by describing a modified version of HMOSAIC [18] called Action-based Language (ABL), the hierarchical version of a theory of motor control development (MOSAIC [19]), to the extent of adding a neural circuitry to process the speech within action predictor and controller modules (Figure 2.2). The feedback signals processed for these modules would also be fed to those circuitries. The theory implies that language function is obtained by exploiting the neural circuitry that studies for goal-related action. They support this architecture through some psychological evidences for human brain. For example, according to their observations, the controller for articulating a certain word overlaps with the controller that generates the action related with that word. The observations showed that human infants learn the words easier when they are shown some action primitives that are related with the uttered words. As a conceptual example, the authors provide a simple interaction scheme where a father asks his infant child to give a cup. Not having already grounded the word “give”, the child would only start to

understand his father's request when he observes that the word "give" is associated with a series of arm motions and grasp and release sequences. They further argue and cite that speech controllers are used to prime and activate the appropriate action controllers for action generation. All of these findings suggest that language should be grounded within actions. This theory can be implemented as an architecture for learning for robots.

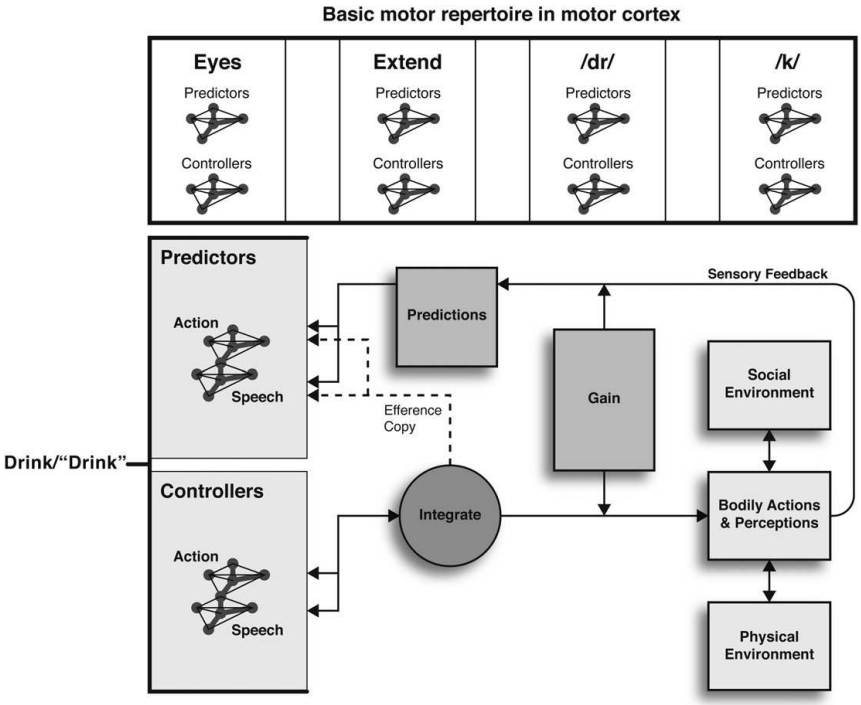


Figure 2.2: The ABL model of [2], exemplifying how a verb (*Drink* in this context) can be understood.

2.2 Concepts

In this section, we summarize the studies and the approaches related to concepts.

2.2.1 Theories of Concept

There are three main views on how concepts can be learned or represented [20, 21, 22, 23]:

- **The Classical, or Rule-based, View**:

  In this view (see, e.g., [24]), categories are exact with strict boundaries; i.e., an examplar is a member of a category or not a member of a category; there is no vagueness involved.

The members of the category share common properties (like YELLOW as colour and LONG as appearance) and the membership for the category is based on satisfying the common properties of the category, established as rules (like *colour of examplar* = YELLOW $\wedge$ *appearance of examplar* = LONG).

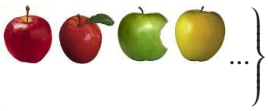
- **The Prototype-based View**:

  In this view, the membership for the categories is confidence-based (e.g., [22]) and the boundaries are not tight. Categories are represented by "prototype" stimuli (the stimuli best representing the category), which are used for judging the membership of other items. The representation of the prototype is mostly based on statistical regularities (i.e., the frequency distribution of the features) [25]. For example, the APPLE concept can be represented by:

$$\text{APPLE} = \begin{cases} \textit{colour} & \text{RED, YELLOW or GREEN respectively} \\ & \text{50\%, 25\% and 25\% of the cases} \\ \textit{shape} & \\ \ldots & \end{cases}$$

- **The Exemplar-based View**:

  In this view, concepts are represented by the exemplars of the category stored in memory (e.g., [26]). An item is classified as a member of a category if it is similar to one of the stored exemplars of the category. For example, the APPLE concept can be represented by:

APPLE = 

  Although the exemplar-based view is in accordance with some experimental results, it falls short in explaining several findings (see [20] for a review and discussion).

Although it is widely believed that the classical view is not adopted by human cognition, there are contradicting evidences about whether humans use prototypes, exemplars or rules for representing concepts [27, 28, 29]. It might be even that for different tasks (e.g., inferencing or classification), we might be using different types of representations [30], making a hybrid representation appealing [31]. Overall, how we represent concepts is still an open issue [32, 33].

2.2.2 Symbol Grounding, Concepts and Language

Perhaps one of the biggest challenge in artificial intelligence is the symbol grounding problem [5]. Harnad [34] argued that the gap between the symbols (i.e., words) and their meanings cannot be closed by an external programmer and that the symbols should be grounded in the *sensory projections* of the objects and the events in the environment. Otherwise, it would be like, as he puts it, learning Chinese from the Chinese dictionary. These observations are critical in the sense that there are uncountable number of applications that robots are expected to undertake, each of which would require an authentic understanding and formulation of the environment. Addressing the establishment of required symbolizations by external means is hence not only intractable, but also meaningless. Although Harnad's approach to intelligence as a *symbol grounding problem* has initiated a great deal of debate, it received a lot of support from the community [35, 36, 37, 38]. Now, it is widely accepted that language, for example, should be grounded in the sensorimotor experiences of the organism [39, 40, 41, 42] (see [43] for a review), and that processing of a word triggers or requires the neural circuitry in the brain corresponding to its sensorimotor experience, meaning or simulation [44, 45] (see [37] for a review); in other words, comprehension of words is likely to involve or require the simulation of the meaning represented by the corresponding concept.

Regarding this problem, there are further clues discovered in the field of neuroscience. Rizzolatti et al.'s discovery of mirron neuron system is such an example [46]. Located on the F5 area of the monkey brain, this group of neurons act both when an action is generated and observed. Like with the Action-Based Language architecture [2] discussed in Section 2.1, this finding of mirror neuron is extensively studied and utilized. Some go further to argue that this is an evidence for the "collective consciousness" [47], but there is at least a consensus that mirror neurons serve to construct a shared grounding for sensorimotor representations [48, 49]. Still, the way our own brains resolve the symbol grounding problem remains as a mystery to be resolved.

The symbol grounding problem in the scope of noun learning has been studied by many. For example, Yu and Ballard [50] proposed a system that collects sequences of images alongside speech. After speech processing and object detection, objects and nouns inside the given speech are related using a generative correspondence model. Carbonetto et al. [51] presented a system that splits a given image into regions and finds a proper mapping between regions

and nouns inside the given dictionary using a probabilistic translation mode similar to a machine translation problem. On another side, Saunders et al. [52] suggested an interactive approach to learn lexical semantics by demonstrating how an agent can use heuristics to learn simple shapes which are presented by a tutor with unrestricted speech. Their method matches perceptual changes in robot's sensors with the spoken words and trains k-nearest neighbor algorithm in order to learn the names of shapes. In similar studies, Cangelosi et al. [43, 53] use neural networks to link words with behaviours of robots and the extracted visual features. Nolfi [54] used the tactile information retrieved from touch sensors for categorizing objects. Similarly, Sinapov [55] categorized objects using proprioceptive as well as auditory signals that the robot experiences while interacting with the objects. In a prior study in KOVAN research laboratory [56], which we extend in this thesis, a robot (iCub) explored the affordances of the objects with a fixed repertoire of behaviors and then, objects were grouped based on what they afforded.

As for the 'verb concepts', the literature has attributed verbs to individual behaviors [57, 58, 59] without generalization considerations. Similar to us, Rudolph et al. [60] proposed relating behaviors to their effects although for a different purpose. They suggested that behaviors be represented in terms of their effects, i.e., changes in features extracted from the environment prior to and after behavior execution. They used their proposal for learning a complex mapping between the hit point and the target point of a thrown ball, and they did not pursue generalization over behaviors or effects nor did they relate their representations to 'verb concepts'. [61] have also studied generalization over behaviors based on their effects in the context of imitation; in their conceptual proposal, they claimed that a robot should use the equivalance of the effects of behaviors for imitating a human performing an behavior rather than performing geometrical transformations between the different embodiments.

Since concepts and language are tightly linked, we also mention studies directly relating words (i.e., symbols) to the sensorimotor data and refer to [2, 43, 62, 63, 64] for reviews on other computational efforts as well as the importance of action and perception in the development of language, and how and why language should be grounded in action and perception.

Cangelosi et al. [43] presents a review of their earlier study (all using multi-layer neural networks) on (i) the multi-agent modeling of grounding and language development, using simulated agents that discover labels, or words, for edible and non-edible food while navigating in

a limited environment [53], (ii) the transfer of symbol grounding, using one simulated teacher (agent) and one simulated learner (agent) that learn new behaviors based on the symbolic representations of the previously learned behaviors [65] and (iii) language comprehension in a humanoid robot, where the robot learns to associate words with its behaviors and the objects in the environment. Similarly, in an earlier study, Cangelosi and Parisi [66] use a neural network for linking nouns to two different objects (a vertical bar and a horizontal bar) and verbs to two different behaviors (pushing and pulling).

Another study on linking language with sensorimotor data [67] demonstrates emergence of symbols (to be linked with language) by interpreting the attractors of a dynamical system (namely, a chaotic neural network) to different symbols and the transitions between the attractors to symbolic manipulation. For a similar goal, Steels et al. [68] demonstrates (using a robot and software simulation) the Recruitment Theory of Language, which claims that organisms try different cognitive or motor abilities for communication first and adapt and develop those that lead to successful communication.

As mentioned by Nehaniv et al. [62], although there exist computational modeling efforts for the emergence of symbols or words for nouns, the emergence of the symbolic representations for verbs is still mostly untouched (except for a study of Cangelosi [43]). Moreover, although highly promising, the efforts for grounding nouns and verbs mostly do not tackle the issues of generalization over entities or behaviors for representing the concepts or the symbols (e.g., [39, 53, 66]), which is, in fact, the most essential reason for having concepts in a cognitive system.

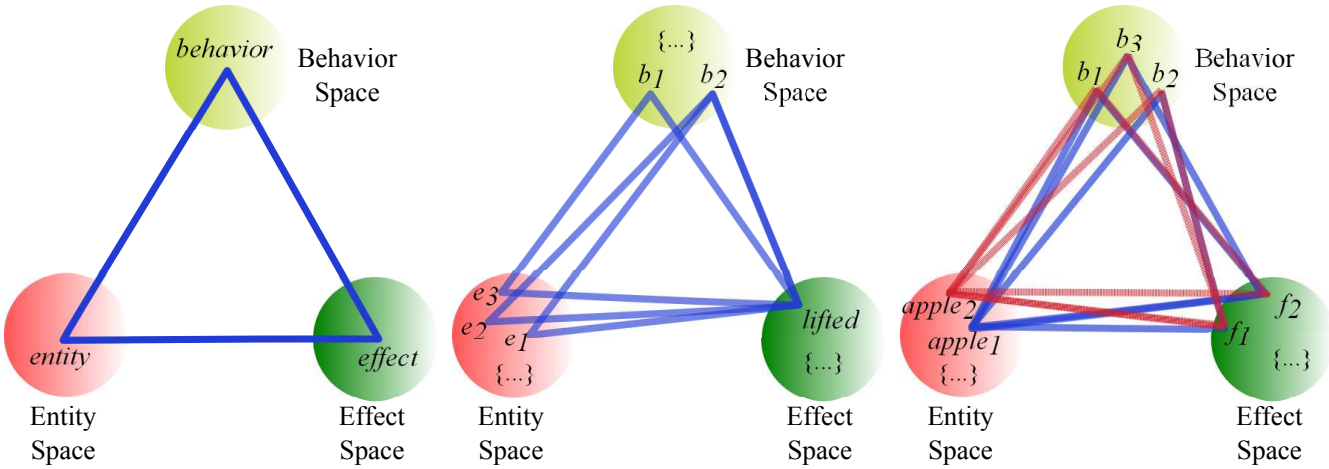
2.3 Affordances

Affordances, a concept introduced by J. J. Gibson [9], offers a promising solution towards symbol grounding since it ties perception, action and language naturally. J. J. Gibson defined affordances as the action possibilities offered by objects to an agent: Firstly, he argued that organisms infer possible actions that can be applied on a certain object directly and without any mental calculation. In addition, he stated that, while organisms process such possible actions, they only take into account relevant perceptual data, which is called as perceptual economy. Finally, Gibson indicated that affordances are relative, and it is neither defined by

the habitat nor by the organism alone but through their interactions with the habitat.

Based on Gibson's ideas and observations, Şahin et al. [11] formalized affordances as a triplet (see, e.g., [69, 70, 71] for similar formalizations):

$$(o, b, f), \tag{2.1}$$



(a) A one-to-one mapping over entity, behavior and effect spaces (b) Generalization of relations over effects (c) Generalization of relation over entities

Figure 2.3: The depiction of the affordance formalization framework. The verb concepts can be obtained via generalizations of affordance relations over the effects (as in Figure 2.3(b) and the noun and adjectives concepts can be obtained via generalizations over entities (as in Figure 2.3(c)).

For example, a behaviour b_{lift} that produces an effect f_{lifted} on an object e_{cup} forms an affordance relation $(e_{cup}, b_{lift}, f_{lifted})$. Throughout this thesis, we use the words object and entity interchangably. Note that an agent would require more of such relations on different objects and behaviours to learn more general affordance relations.

During the last decade, this and similar formalization approach of affordances proved to be very practical with successful applications to domains such as navigation [72, 73], manipulation [3, 74, 75, 76, 77], conceptualization and language [78, 79], planning [3], imitation and emulation [3, 69, 79], tool use [70, 80, 81] and vision [79]. A notable one with a notion of affordances similar to ours is presented by Montesano et al. [82, 83]. Using the data obtained from the interactions with the environment, they construct a Bayesian network where the correlations between actions, entities and effects are probabilistically mapped. Such an architecture allows action, entity and effect information to be separately queried (given the

other two information) and used in various tasks, such as goal emulation.

The usage of affordances are reported to be extensible towards taking account of human-robot interactions. In some related studies, affordances (called "interpersonal affordances") that emerge from coordinated joint actions of two robots are investigated [84, 85]; e.g., two robots learning the interpersonal affordance of lifting a Table which, otherwise, is not liftable by either of them. This study significant in the sense that the human is seen as part of the environment (with no special status) and uses the very same framework to learn social affordances as the physical affordances of objects.

2.4 Curse of Dimensionality and Feature Selection

One of the biggest challenges in analyzing data in machine learning, data mining etc. is the curse of dimensionality, first pointed out by Richard Bellman in his book "*Adaptive control processes: A guided tour*" [86]. In short, it is the problem that arise when the number of feature dimensions increase and cause the data points to become sparse. As such, the comparison and clustering of data will yield undesired similarities. To be able to produce meaningful groupings of data, a good design approach must handle this phenomenon.

Many attempted to solve this problem e.g. by exploring on various exploitations the density information [87], applying signal processing techniques like wavelet transform [88], and so forth. Steinbach et al. [89] thoroughly review the curse of dimensionality and the attempt to overcome this problem in the domain of document classifications. Incidentally, they mention about *concept based clustering* in his study on document classifications, where the feature vectors are only used to form some high-level conceptual information and the clustering is performed on the so-called concept space. In contrast to our view, their notion of *concepts* is dominated with rather non-linguistic, domain related tags such as *Art* or *Finance*, and their approach seems to be forming a higher level of feature vectors. In another study [16], a version of Slow Feature Analysis (SFA) is used to detect the visual features that slowly change over time and through this process, some few abstract spatio-temporal are obtained. The methodology presented here is very practical for the robotic applications where the robot's environment is endowed with a continuous stream of its interactions with objects and people.

Dividing the feature set into logical subsets is another approach to handle the curse of di-

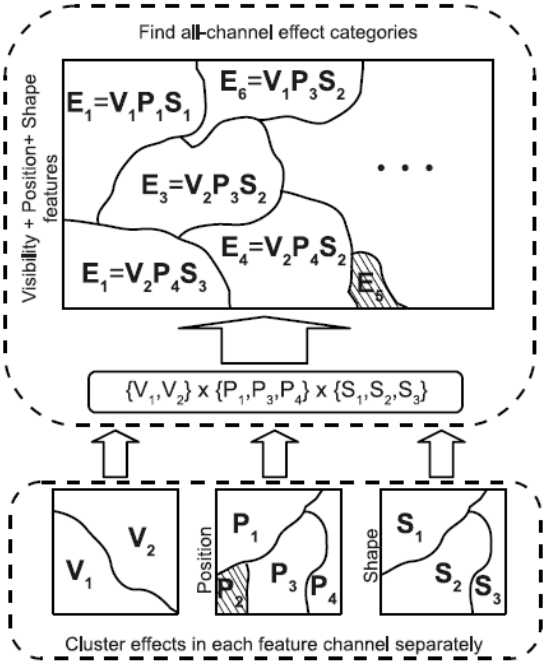


Figure 2.4: The hierarchical clustering described by Ugur et al. [3], which groups the information derived from interacting with the environment into effect categories. In this study, the effect categories that have small sizes are discarded automatically.

mensionality. These subsets would be separately analyzed for clustering and in most cases play role in a stage of hierarchical clustering. For robotics applications, this logical subsets might correspond to different sensor modalities. One good example is in presented by Haazebroek et al. [1], where with the remark that primate brain processes different features across different cortical maps, different perceptual modalities and even different dimensions are clustered and represented in different sensory maps. As another instance, [4] separates the data into four subsets, and processed them accordingly: shape, color, spoken words, and robot's posture (see Figure 2.6). In a similar manner, [15] apply clustering on two subsets of data, namely proprioceptive and auditory feedback. As a very relevant that adopts the same computational framework of affordances as we do, [3] applies a hierarchical clustering algorithm that first clusters the data within three feature channels (visibility, distance, shape) using X-means algorithm [90] and then takes the cartesian product of these clusters to obtain all-channel clusters (see Figure 2.4). In most cases of such methodologies, one must also define a measure to handle the fusion of these clusters of feature subsets, and in some cases, to discard the irrelevant cluster combinations.

As an alternative approach, there are some discrimination methods to analyze and preprocess the data into a better separable state, usually by weighting or eliminating features. Kira and Rendell's relief [91] and its extension (reliefF) by Kononenko [92] are the two of the some successful instances to accomplish this task, where the features are weighted according to their discriminative power. These algorithms are useful also in the sense that they can be applied to cases where some features show dependency to each other, thus, allowing inclusion of redundant features for enhance expressivity.

While some unsupervised feature weighting algorithms are available, literature instances surveyed so far do not have a generic usage. The study of Frigui et.al. [93] concentrates on such an unsupervised feature discrimination methods for image database categorization with dividing the set of features into logical subsets. The designation of such subsets are specific to the image categorization domain and still involves some indirect supervision. Zeng [94] also proposes an unsupervised technique, formulating the feature selection in clustering as an optimization problem.

2.5 Support Vector Machine for Supervised Learning

In his book called *Statistical Learning Theory* [95], Vapnik introduced Support Vector Machines as methods for statistical classification and regression analysis. While originally proposed as a purely linear classifier, as depicted in Figure 2.5, the algorithm can also first map the data into a high-dimensional feature space where it is linearly separable, then find the hyperplanes that optimally separate data with different labels.

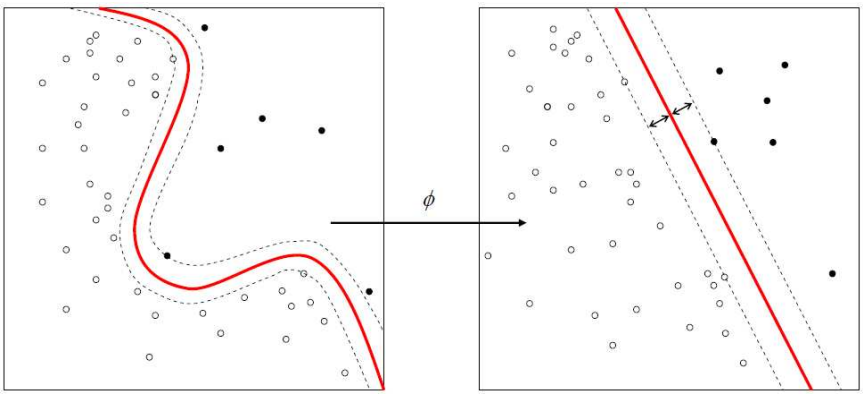


Figure 2.5: Using kernel functions, the space can be transformed such that the mapped data is linearly separable. Then, the optimally separating hyperplane can be calculated.

As a powerful supervised learning technique, SVMs are successfully applied studies related with robotics [3, 7, 8, 56, 73, 79, 78]. In our work, we have chosen to use the SVM implementation of LibSVM [96], an open source library that includes different kernels and options (n-fold cross validation, probability estimates) for Support Vector Machines.

2.6 Self Organizing Maps

A self organizing map (SOM) [97], or Kohonen Map, is an unsupervised neural network technique. It employs a competitive learning among nodes that form a m-by-n grid topology. For every version of self-organizing map, it is typical to feed the input to the map, which results in selection of the node with the closest weight value (winner node) according to some distance function (activation function) and modification of the weight of the said node and its neighbors according to an update function. With this nature, the self-organizing maps are the examples for incremental learning. This simple procedure is repeated with the whole training set for a predetermined number of epochs. Since the initial weights for the nodes are given randomly, Self-Organizing Maps and all other variants are stochastic processes. Therefore, the algorithm may find different clusters at each run, requiring

The map topology, activation function and update functions may vary by design choice. An extension (Growing Self Organizing Map [98]) addressed the automatic determination of the size of the map by tuning a parameter called spread factor. Further algorithms inspired from SOM include Neural Gas algorithms [99].

The variations of Self Organizing Maps are successfully applied within robot learning architectures with a vast variety of features. For example, Cangelosi et al. [4] proposes a model with four interconnecting SOMs that deal with pattern recognition process in auditory, shape, colour, and posture information (Figure 2.6). Sinapov and Stoytchev [15] implement SOMs for proprioceptive and auditory sensor feedback to associate sensory feedback with the applied behavior on various objects. Similarly, Natale et al. [100] utilizes Self-Organizing Maps on tactile sensor information to capture the similarities between various objects.

When analyzing data with multiple number of classes, conventional supervised techniques such as SVM may fail to yield accurate classifications. To overcome this difficult issue, inspiring from the studies in robotics [4, 15, 100], we have chosen to implement SOMs.

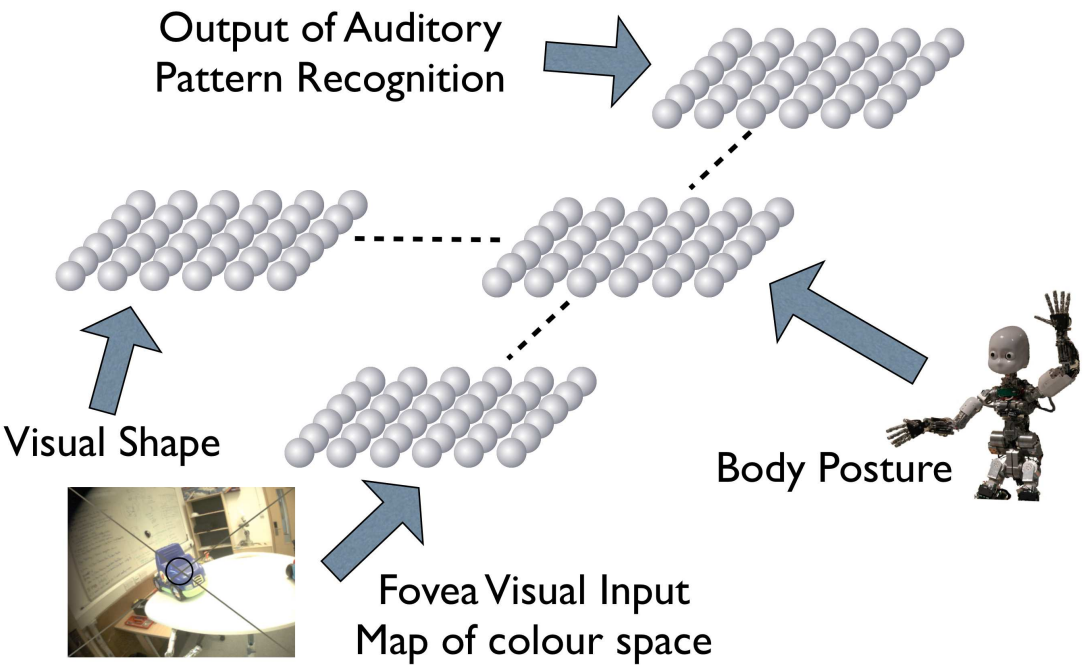


Figure 2.6: The learning architecture proposed by Morse et al. [4]. The grids of grey balls conceptually represent self-organizing maps and each grey ball is a node (unit).

CHAPTER 3

SETUP AND METHODS

In this chapter, we explain the methodological details on our system and the underlying setup.

3.1 Setup

In this section, we describe the hardware and software platforms used in the thesis.

3.1.1 The iCub Humanoid Platform

iCub [101] is a fully open source humanoid robot designed for cognitive and developmental robotics research (see Figure 3.1). It has the dimensions of a 3.5 years of child, and has 53 degrees of freedom distributed on its legs, torso, and (mostly) arm and hands. While there are different versions of this platform, ours is endowed with force and torque sensors on its arms and tactile sensors on its hands.

3.1.2 Behaviors

The robot's behaviour repertoire $\mathcal{B}$ contains six behaviors ($b_1, ..., b_6$ - Eq. 2.1): *top-grasp*, *side-grasp*, *push-left*, *push-right*, *push-forward*, *pull*. These behaviors were implemented using the inverse kinematics and action primitives libraries available within the iCub software repository. In pushing behaviors, the hand of iCub that is closer to the object is selected and moved towards the object, and then 20 centimeters towards the desired direction. In grasping behaviors, iCub moves its hand towards the object until it hits the object (the hit is

perceived according to the tactile and force feedback), then closes the hand. Note that, in a fully developmental learning architecture, the behavioral parameters are also learned.

The behaviors are designed and implemented in a simplistic manner for the sake of the scope of this thesis. Note that our experiments only covers a *subset* of possible behaviors. The implications of this choice is briefly discussed in Chapter 5.

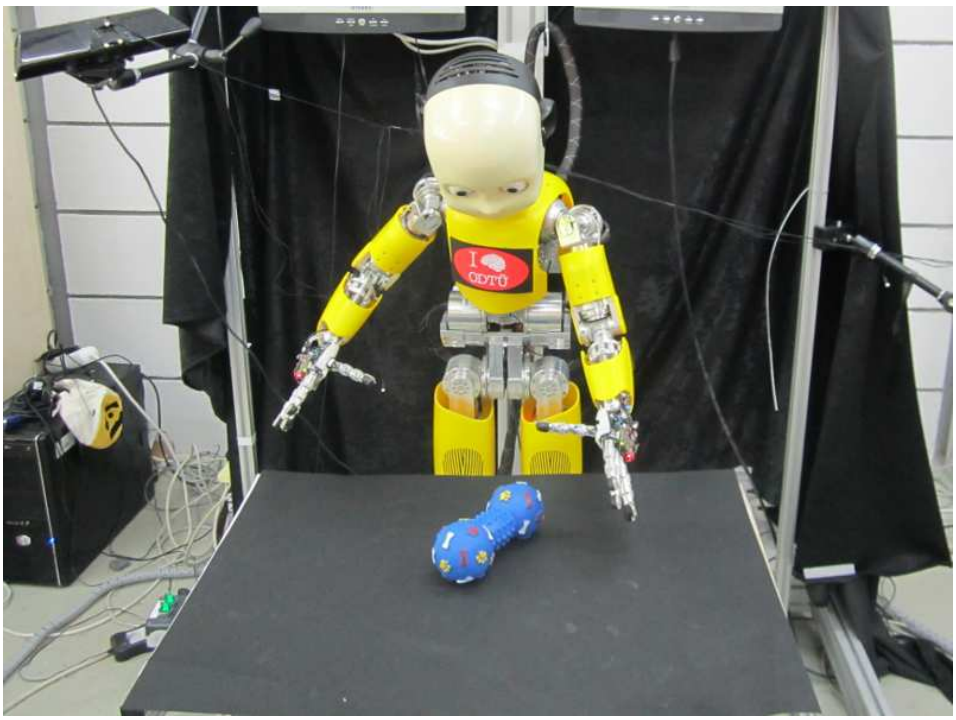


Figure 3.1: The iCub Humanoid platform with a Kinect sensor at side.

3.1.3 Yet Another Robotic Platform (YARP), Robotic Operating System (ROS) and iCub Software

We use YARP [102, 103] to set up and maintain a network between the parts of iCub, i.e. arms, legs, head, torso (See Appendix B.1 for a brief overview of YARP). We have used the software provided by iCub community which utilizes this network infrasturcture and contains motion control modules for arm, gaze and grasp.

We use ROS [104] to implement a service-oriented system which has nodes that manage perception, interface with iCub motion control and experiment routine (See Appendix B.2 for a brief overview for ROS).

3.1.4 Perception

For perceiving the environment and establishing coordination with the motion control, we make use of various hardware and software platforms, as discussed in the following sections.

3.1.4.1 Kinect

iCub perceives the environment with a Kinect sensor (Figures 3.1 and 3.2). Originally developped for a commercial video game console, it is now also widely used within the robotics and computer vision research. It features a depth sensor and an RGB camera; both with 640x480 resolution and 30 Hertz frame rate. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor. It is observed that Kinect performs best in capturing visual data within 1 to 4 meters.

Figure 3.2: The Kinect sensor.

3.1.4.2 VisualEyez VZ4000

VisualEyez VZ4000 (Figure 3.3) is a high performance real-time 3D motion capture system. It uses tiny LEDs as markers, providing a very good approximation for defining points in the 3D space with practically no partial occlusion. The device can capture motion within a volume of 190 m^3. While some of the works held in our lab employs this use case, our work simply uses this hardware for calibrating the coordinate axes of Kinect and iCub. To do so, we place three LEDs on the coordinate frame of iCub and execute the calibration algorithm

Figure 3.3: The VisualEyez VZ4000 tracking device.

provided within the proprietary software for VZ4000.

3.1.4.3 Features for Objects and Effects

In order to simplify perceptual processing, we assumed that iCub's interaction workspace is dominated by an interaction table. In this thesis, we have employed curvature and normal esimation methods provided by PCL, an open source library [105]. The table is assumed to be planar and is segmented out as background.

Perception process starts with segmenting tabletop-object from raw point cloud input (Figure 3.4(b)). The segments are fit with (PCA based) oriented bounding boxes (Figure 3.4(c)) and identified (Figure 3.4(d)) so that the sensory information about objects can be separately queried by other software modules (learning, visualization, planning etc.). Bounding box identification algorithm checks the volumetric intersection between two consecutive frames, and checks the dimension-wise similarity between an unidentified box and the boxes from previous frame or the boxes that have already been identified until that particular frame.

Boxes are also filtered out if they are bigger than a certain threshold which might indicate a case where a human or robot hand comes into the scene. At the end, an entity (object) is represented by an oriented bounding box and a point-cloud inside this box.

After these processes, the following features are extracted to represent an object o (Eq. 2.1):

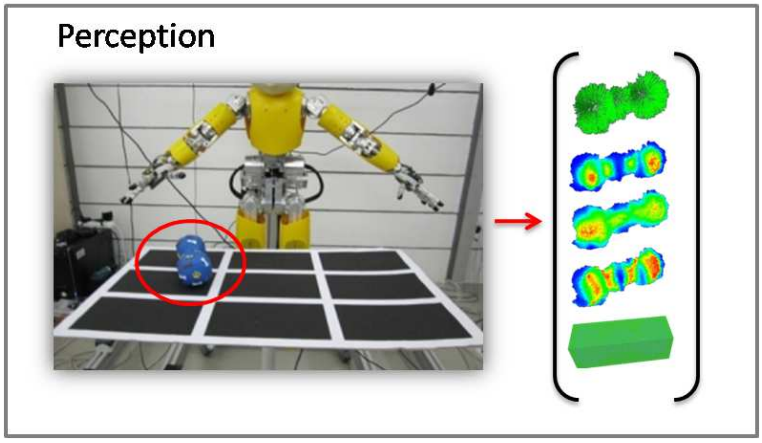
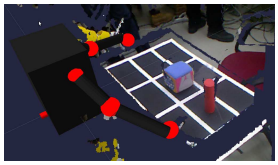
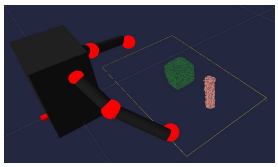


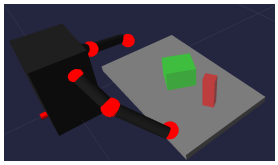
Figure 3.4: Snapshots from *rviz* ROS visualization tool. The perception system efficiently detects and displays the objects on the table. iCub is represented by primitive shapes which becomes useful if the 3D points corresponding to the robot body is to be filtered out.



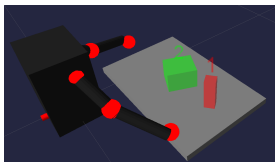
(a) raw



(b) clustered



(c) oriented bounding boxed



(d) identified

- *Surface features:* surface normals (azimuth and zenith angles), principal curvatures (min and max), and shape index. They are represented as a 20-bin histogram in addition to the minimum, maximum, mean, standard deviation and variance information.

- *Spatial features:* bounding box pose (x, y, z, theta), bounding box dimensions (x, y, z), and object presence.

- *Object Presence:* A binary feature for whether an object exists in the scene. This information is especially useful when an object disappears after an interaction.

Note that there are both high-level and low-level features in our set.

For the effect information, we take the difference between the feature vectors of an entity before and after the behavior is applied.

Furthermore, we have added the language feature. This is a discrete feature that is the enumeration that ranges within the number of uniquely observed effect. See Section 3.4.1 for

further explanation on the utilization of this feature.

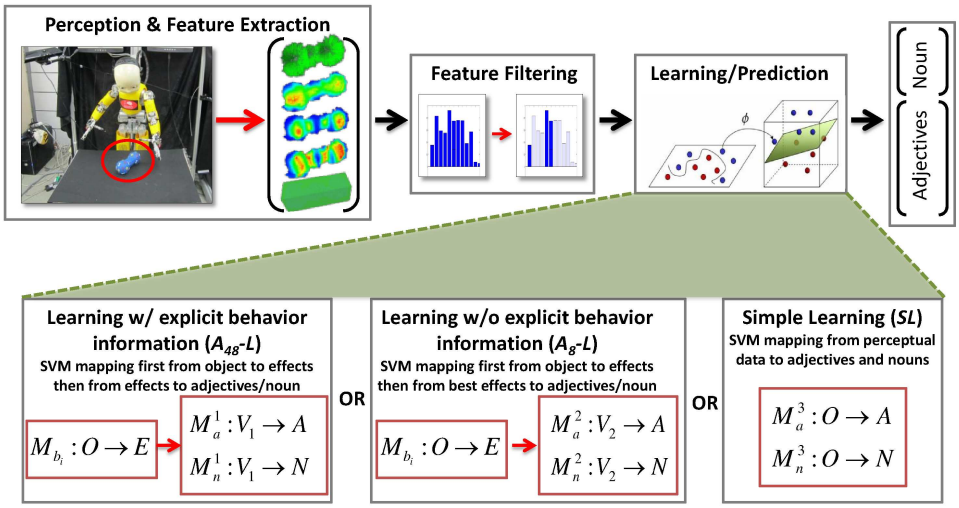
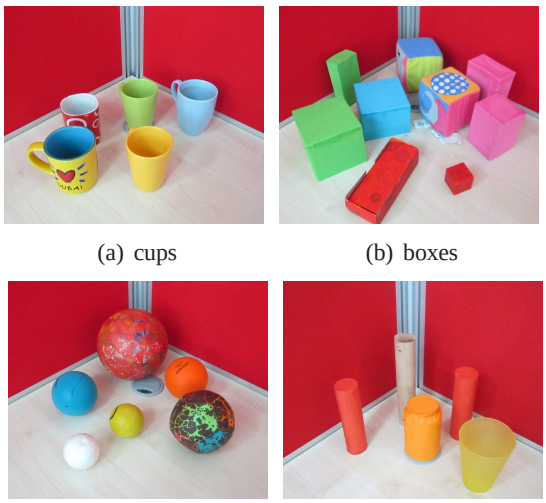


Figure 3.5: Overview of the system. iCub perceives the environment and learns the affordances. From the affordances learned from the perceptual data, adjective classifiers are learned.

3.2 Data Collection

The robot interacted with a set of 35 objects of variable shapes and sizes, which are assigned the nouns “cylinder”, “ball”, “cup”, “box” (Fig. 3.6).



(a) cups (b) boxes

(c) balls (d) cylinders

Figure 3.6: The objects in our dataset.

iCub applies each behaviour b_j on each object o_i and observes an effect $f_{o_i}^{b_j} = o_i' - o_i$, where

o'_i is the set of features extracted from the object after behaviour b_j is applied. After each interaction epoch, we give an appropriate effect label $E_k \in \mathcal{E}$ to the observed effect $f_{o_i}^{b_j}$, where E can take values `Moved Left`, `Moved Right`, `Moved Forward`, `Pulled`, `Grasped`, `Knocked`, `Disappeared` or `No change`[1]. Thus, we have a collection of $\{o_i, b_j, E_{o_i}^{b_j}\}$, including an effect label $E_{o_i}^{b_j}$ for the effect of applying each behaviour b_j to each object o_i.

3.3 Affordance Learning

Using the effect labels $E \in \mathcal{E}$, we train a Support Vector Machine (SVM) classifier for each behavior b_i to learn a mapping $\mathcal{M}_{b_i} : \mathcal{O} \rightarrow \mathcal{E}$ from the initial representation of the objects (i.e., $\mathcal{O}$) to the effect labels ($\mathcal{E}$). The trained SVMs can be then used to predict the effect (label) $E_{o_l}^{b_k}$ of a behavior b_k on a novel object o_l using the trained mapping $\mathcal{M}_{b_k}$. Before training SVMs, we use ReliefF feature selection algorithm [91] and only use the features with important contribution (weight > 0) to training.

3.4 Category Learning

In this section, we discuss inferring categories using affordances.

3.4.1 Verb Learning Model

In this section, we explain how the self-organizing map algorithm is modified and utilized to meet our needs on learning concepts. We present the ground truth for effects as a separate feature, namely, the *language feature*. With this feature, our approach depicts a learning model similar to those of Glenberg [2] and Cangelosi et al. [4]. Since SOM algorithm is a stochastic process, we run it with multiple trials on the same input. Then, all trials are compared with the ground truth. The best trial is passed on to the further steps.

To have a quick convergence, all units are initialized with a unique value in the language feature, which ranges within the labels given for effects (1 for *no change*, 2 for *moved left*,

[1] The *no-change* label means that the applied behavior could not generate any notable change on the object. For example, iCub cannot properly grasp objects larger than its hand, hence, the *grasp* behaviour on large objects do not generate any change.

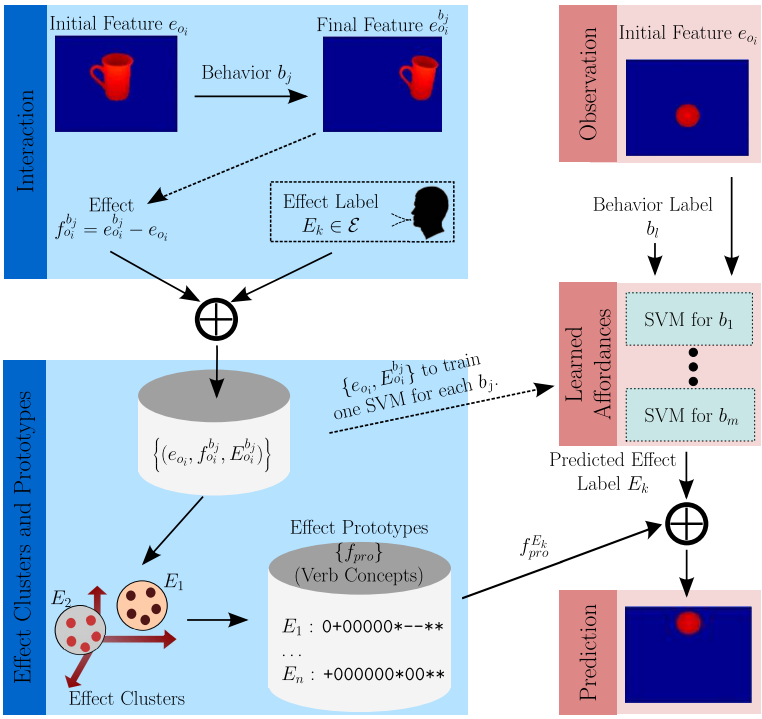


Figure 3.7: The mechanism for affordance learning. We collect instances of effects and give them effect labels. These labels are in turn used to train SVMs, and separately, forming effect clusters.

and so forth). Methodologies similar to this are sometimes referred as *priming* in related literature [106, 107, 108].

Considering that different activation and update functions can yield different performances, we have tested the following alternative function sets:

- *Type-I*: A straightforward approach would be with an initial activation function as:

$$A = \sqrt{\sum_{i=1}^{n}(v - w_i)}, \tag{3.1}$$

where A refers to the resulting activity of each node in the map, v is the input vector, and w_i is the weight vector of node i. The node with the smallest A_i is considered to be the winning node given v. When the winner node is determined, the final activation function is used to determine a multiplier to be used in weight update:

$$y_i = exp\left(\frac{-\beta_i}{2\sqrt{n}}\right), \tag{3.2}$$

where y_i is the final activation of ith node in the map, β_i is the distance from node i to the winning unit, and n is the total number of nodes in the map.

- *Type-II*: The second alternative uses the standard Euclidian distance for the initial activation function:

$$A = \sqrt{\sum_{i=1}^{n}(v - w_i)^2}. \tag{3.3}$$

The variables are as explained in Equation 3.1. The final activation is also modified as

$$y_i = exp\left(\frac{-\beta_i^2}{s_n^2}\right), \tag{3.4}$$

where s_n is the neighbourhood size, and the other variables are the same with Equation 3.2.

- *Type-III*: The last alternative employs Minkowski distance in calculating the winner node with the following equation:

$$A = \left(\sum_{i=1}^{n}|v - w_i|^f\right)^{\frac{1}{f}}, \tag{3.5}$$

where f is the number of features in the feature vector. The final activation is as in the Equation 3.4

Using one of the final activation equations (Equation 3.2 or Equation 3.4), we obtain y_i and change the weight of each node with the following equation:

$$\Delta w_{ij} = \alpha(v_i - w_{ij})y_i, \tag{3.6}$$

where α is the learning rate, empirically set to 0.1, and w_{ij} is the weight of ith node for j^{th} feature.

In computing the distance between language and object presence features, we apply the discrete distance, i.e., for a given input vector v and the weight vector w_i of node i,

$$d(v_{language}, w_i^{language}) = \begin{cases} 1 & v_{language} \neq w_{language} \\ 0 & otherwise \end{cases}, \tag{3.7}$$

where $v_{language}$ and $w_i^{language}$ are the language values. Furthermore, the winner update is modified accordingly. Each node in the SOM keeps record of number of instances for each of the different language enumerations. When the node i becomes a winner, the language value in its weight vector is updated as follows:

$$w^i_{language} = \arg\max_l count(v_{language} = l). \tag{3.8}$$

This approach might be criticized in the sense that:

- The enumeration we used in our experiments for the language information do not depict the similarity between effects.
- The advent of neighbourhood of nodes can be discarded.

Rather than directly giving the labels, the language features can be provided as outputs from any given state-of-the-art speech processing systems. This speech processing extension is specifically excluded for the sake of preserving the focus on scope.

As to demonstrate the usefulness of language feature in forming effect clusters, we have also run the SOM algorithm on the dataset without the language feature. Using all of the options presented in *Type-I*,*Type-II* and *Type-III* SOM, we take the highest prediction rate obtained without the language feature. We abbreviate this type of SOM as *Type-0* SOM. The results are briefly compared in Section 4.2.1.

3.4.2 Adjective Learning Model

We train SVMs for learning the adjectives of objects from their affordances (see Fig. 3.5). We have six adjectives, i.e., $\mathcal{A}$ = {'*edgy*'-'*round*', '*short*'-'*tall*', '*thin*'-'*thick*'}, for which we require three SVMs (one for each pair). We have the following two alternative adjective learning models:

- *Adjective learning with explicit behavior information (**A**$_{48}$-**AL**)*:
  In the first adjective learning model, for learning adjectives $a \in \mathcal{A}$, we use the trained SVMs for affordances (i.e., $\mathcal{M}_b$ in Sect. 2.3) to acquire a **48-dimensional** space, $\mathcal{V}_1$ =

$(\hat{E}_1^{b_1}, ..., \hat{E}_8^{b_1}, ..., \hat{E}_1^{b_6}, ..., \hat{E}_8^{b_6})$, where $\hat{E}_i^{b_j}$ is the confidence of behaviour b_j producing effect E_i on the object o. We train an SVM for learning the mapping $\mathcal{M}_a^1 : \mathcal{V}_1 \rightarrow \mathcal{A}$ (See Figure 3.8 for the detailed sketch of this model).

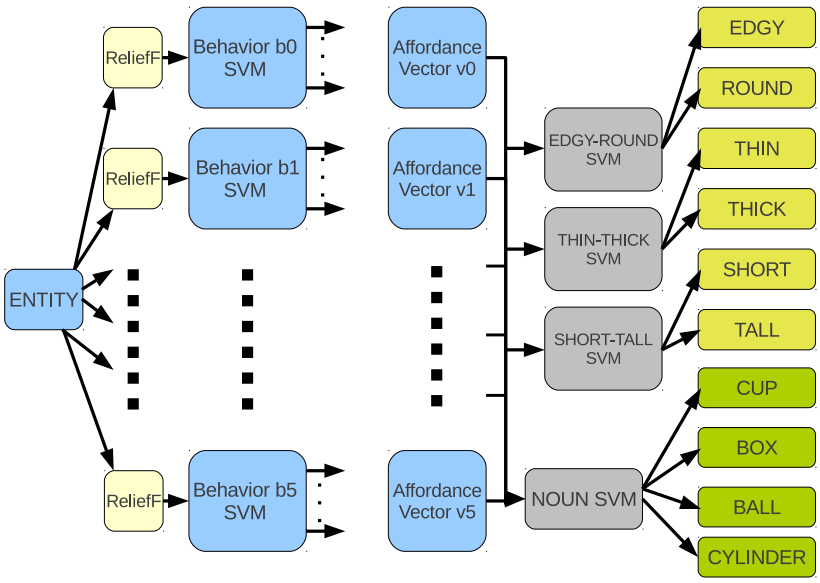


Figure 3.8: The mechanism with explicit behavior information for adjective and noun learning. Each adjective and noun SVM classifier is fed with 48 dimensional affordance vectors, which is a concatenation of 8-dimensional affordance estimate vectors obtained from each behaviorwise SVM.

- *Adjective learning without explicit behavior information (**A**$_8$-**AL**)*:

  In the second adjective learning model, for learning adjectives $a \in \mathcal{A}$, we use the trained SVMs for affordances to acquire an **8-dimensional** affordance vector, $\mathcal{V}_2 = (p(E_1), ..., p(E_8))$, where $p(E_i)$ is the maximum SVM confidence of a behaviour b_j leading to the effect E_i on object o. From $\mathcal{V}_2$, we train an SVM for learning the mapping $\mathcal{M}_a^2 : \mathcal{V}_2 \rightarrow \mathcal{A}$ (See Figure 3.9 for the detailed sketch of this model).

For the sake of comprehending the difference between these two models, consider the following example: Assume that the robot has only two behaviors, namely, *top-grasp* and *side-grasp* (b_1 and b_2 of our behavior repertoire). Consider a set of objects that contains instances that (a) can not be grasped by either behavior, (b) can be grasped by applying either b_1 or b_2, (c) can only be grasped by applying b_1 and (d) can only be grasped by applying b_2 (the dataset

used in our experiments contains such instances). The $\mathbf{A}_{48}$**-AL** model, being a more discriminative approach, can capture such differences. Hence, theoretically speaking, the first model provides a more specialized classification of objects.

After learning, iCub can predict the noun and adjective labels for a novel object (Fig. 4.2).

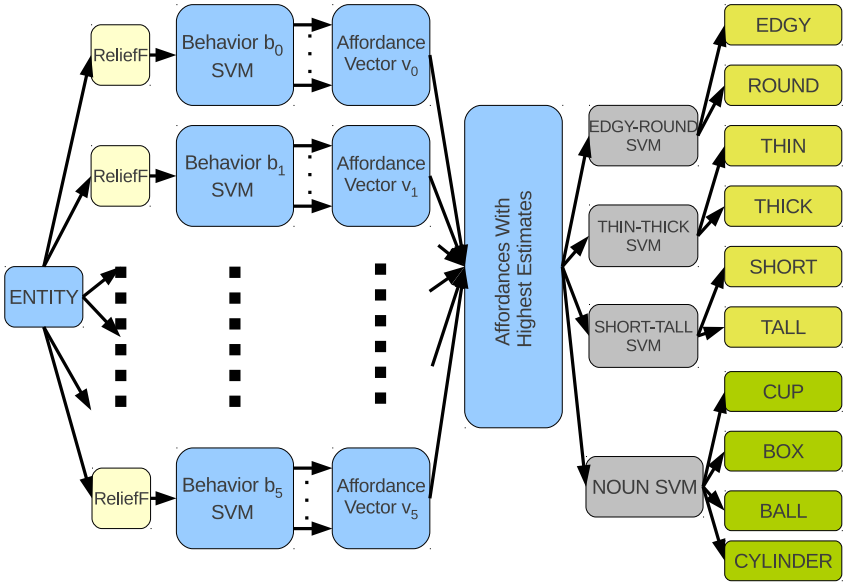


Figure 3.9: The mechanism with no explicit behavior information for adjective and noun learning. Each adjective and noun SVM classifier is fed with 8 dimensional affordance vectors, where each dimension corresponds to the highest estimate obtainable from the given behaviorwise SVM. For example, If the highest estimate for a given object from Push-left SVM is 0.95 (of *moved left* effect), then the corresponding dimension in the affordance vector would be set to 0.95.

3.4.3 Noun Learning Model

We train **one SVM** for nouns $\mathcal{N}$ = {'ball', 'cylinder', 'box', 'cup'}, for which we have 413 instances. Similar to adjectives, we have two models:

- *Noun learning with explicit behavior information (**A**$_{48}$**-NL**)*:
  Similar to A_{48}-AL, we train an SVM for learning the mapping $\mathcal{M}_n^1 : \mathcal{V}_1 \rightarrow \mathcal{N}$ (See Figure 3.8 for a sketch of this model).

- *Noun learning without explicit behavior information (**A**$_8$**-NL**)*:
  Similar to A_8-AL, we train an SVM for learning the mapping $\mathcal{M}_n^2 : \mathcal{V}_2 \rightarrow \mathcal{N}$ (See

Figure 3.9 for a sketch of this model).

3.4.4 Conceptualization

In this section, we discuss obtaining string representations of noun, adjective and verb concepts. These string representations offer a way to distinguish different nouns, adjectives and verbs. See Algorithm 1 for a brief explanation of the procedure applied to obtain string representations for verb concepts. Similar procedure is applied for extracting adjective and noun strings with small modifications: For adjectives, the mean-variance space of affordance confidence values is fed into the algorithm. For noun strings, the entity (object) features are used for the same means.

Algorithm 1 Derivation of String Representations

Require: The labels have been given for effects.

for all E in the set of effect clusters $\mathcal{E}$ **do**

- Compute the mean ${}_i\mu_E$ of the change in each feature element i:

$$ {}_i\mu_E = \frac{1}{N} \sum_{f \in E} {}_i f, \tag{3.9} $$

where N is the cardinality of the set $\{f \in E\}$.

- Compute the variance ${}_i\sigma_E$ of the change in each feature element i:

$$ {}_i\sigma_E = \frac{1}{N} \sum_{f \in E} ({}_i f - {}_i \mu_E)^2. \tag{3.10} $$

end for

- Apply Robust Neural Growing Gas (RGNG) algorithm [109] in the space of $\mu \times \sigma$.

if Extracting effect prototypes **then**

- Manually assign the labels '+', '-', '0' and '*' to the four clusters that emerge in the previous step.

else

- Manually assign the labels '+', '0', and '*' to the three clusters that emerge in the previous step.

end if

3.4.4.1 Effect Prototype Extraction

In previous studies of KOVAN laboratory [7, 8], a process was applied to obtain *effect prototypes*, which are essentially the condensed representations of *verb concepts*. After obtaining effect clusters, we compute the mean and variance of each feature in each effect cluster. Then, we apply an unsupervised algorithm called Robust Growing Neural Gas [109] (see Appendix

A) on the $\mu \times \sigma^2$ space to find out the distribution of these feature changes. We observe that these values are naturally clustered into four categories, which we inspect and comment that they are either *unchanged* (mean close to zero, small variance); *consistently increased* (positive mean, small variance), *consistently decreased* (negative mean, small variance); and *inconsistently changed* (large variance). We denote these categories with the strings 0,+,– and $*$, respectively. We call the collection of these strings as the *string representations of effect prototypes*.

3.4.4.2 Adjective Prototypes

In this thesis, in addition to obtaining effect prototypes, we utilize this algorithm on analyzing the dependencies between adjectives and affordances and produce prototype strings for adjectives. To do so, for each adjective model, we first manually assign vectors of affordances described in section 3.4.2 ($\mathcal{V}_1$ when analyzing $\mathbf{A}_{48}$**-AL** and $\mathcal{V}_2$ when analyzing $\mathbf{A}_{48}$**-AL**) into adjective clusters. Then, similar to effect features in effect clusters, the confidence features in adjective clusters are grouped according to their mean and variance values with the RGNG algorithm. In contrast to effect features, we observe that the types of dependence between each adjective and the effects of the behaviors are naturally clustered into three categories, namely, *Consistently Small*, *Constistently Large* and *Highly Variant*. We denote these categories with $-$, $+$ and $*$, respectively. With this application, we also verify the differences between the adjective models A_48-AL and A_8-AL, which we briefly discussed in section 3.4.2. Since the RGNG algorithm is a stochastic process, we run it with multiple trials on the same input. See section 4.3.2 for the results.

3.5 Metrics and Evaluations

Firstly, using ground truth for each instance in our database, we first evaluate the accuracy of the predictions for affordance, noun and adjective learners. Secondly, since the effect labels are represented by effect prototypes, the similarity between an effect instance and the predicted effect prototype is needed and we use a modified version of Mahalanobis distance, which is calculated by taking the mean μ_{E_i} of first effect cluster E_i (if the first E_i is an effect instance, we take the effect instance as μ_{E_i}) and using the second effect cluster's E_j mean μ_{E_j}

and variance σ_{E_j}:

$$d_{mm}(E_i, E_j) = \sqrt{\left(\mu_{E_i} - f^{+,-,0}_{proto,E_j}\right)^T S_j^{-1} \left(\mu_{E_i} - f^{+,-,0}_{proto,E_j}\right)}, \tag{3.11}$$

where S_j is the covariance matrix of the second effect cluster E_j. In computing the Mahalanobis distance, the features marked inconsistent in the prototype are disregarded (denoted by $f^{+,-,0}_{proto,E_i}$ for the effect prototype f_{proto,E_i} of an effect cluster E_i), as those correspond to an unpredictable/inconsistent change in the feature elements. To demonstrate the usefulness of this formulation, we also show the outcome of distance calculation using pure Euclidian and Mahalanobis distances of an effect instance to the closest elements of clusters. We utilize the Euclidean distance as

$$d_e(f_{new}, E_i) = \min \sqrt{\sum_{n=1}^{N} (f^n_{new} - f^n_e)^2}, \tag{3.12}$$

where N is the number of features extracted from the interactions and f are the instances of effect cluster E_i. The pure Mahalanobis distance is the same with our modified distance except that it considers all features:

$$d_{pm}(E_i, E_j) = \sqrt{\left(\mu_{E_i} - f_{proto,E_j}\right)^T S_j^{-1} \left(\mu_{E_i} - f_{proto,E_j}\right)}. \tag{3.13}$$

We use the Equation 3.11 to demonstrate the Prototype-Based view of concepts, and compare it with Exemplar-Based view of concepts by using Equation 3.12 (Section 2.2.1 reviews these views of concept) and demonstrate the efficiency of discarding inconsistent features by comparing with Equation 3.13, which does not discard those features. Table 4.3 conveys our empirical findings regarding this comparisons.

To evaluate our models, we have also trained SVM classifiers that map entity information directly onto adjective and noun labels (See Figure 3.10 for the detailed sketch of these models):

- *Simple adjective learning (**SAL**)*:
  In this adjective learning model, we learn $\mathcal{M}^3_a : \mathcal{O} \rightarrow \mathcal{A}$ directly from the appearance of the objects.

- *Simple noun learning (**SNL**)*:

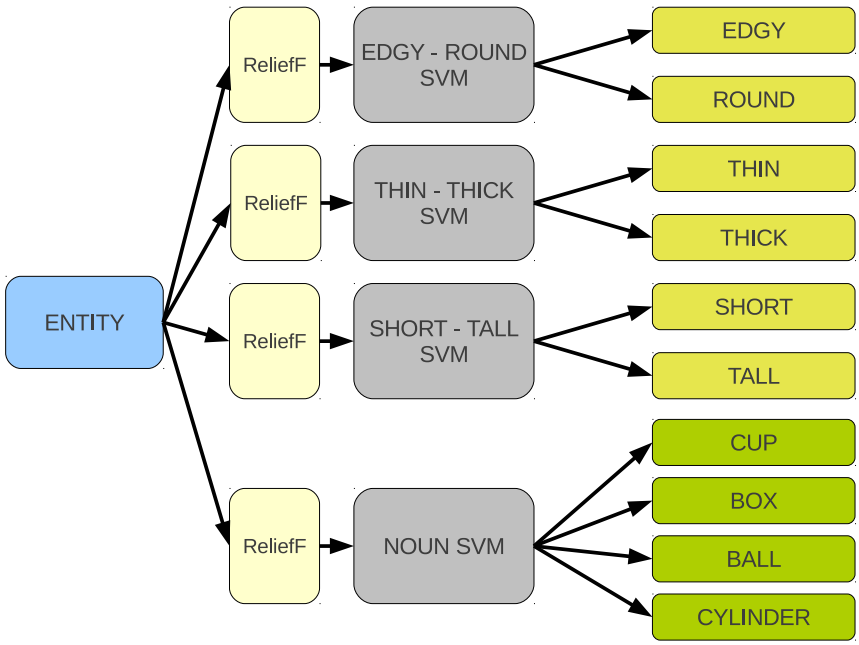


Figure 3.10: The simple noun and adjective learner. Perceptual information are directly fed to the SVM classifiers.

Similar to SAL, we train an SVM for learning the mapping $\mathcal{M}_n^3 : \mathcal{O} \rightarrow \mathcal{N}$ directly from the appearance of the objects.

Furthermore, we trained an SVM classifier (*Simple verb learning (**SVL**)*) that maps effect information directly onto verb labels. Similar with the affordance learning scheme, we we use ReliefF feature selection algorithm and only use the features with important contribution (weight > 0) to training.

3.6 Multi-Step Planning from Verb Concepts

In order to demonstrate the usefulness of concept learning, we experimented on generating composite behavior towards achieving a goal state, that is, multi-step planning. There are some successful works [3, 76] that previously dealt with multi-step planning with forward chaining using affordances. Their approach on applying conceptualized effect information, which is simply adding the mean value of the whole effect vector, corresponds to Exemplar-Based view of concepts. We extend their approach with the advent of effect prototypes as

depicted in Figure 3.11, providing a way to experiment with Prototype-Based view of concepts. Since we trained SVMs for predicting the effect of each behaviour on an object, iCub can do forward chaining to find the set of behaviours leading to a goal state. Since the effect labels are represented by effect prototypes, the similarity between the goal state (which is an effect instance) and the predicted effect prototype is needed and we use The Mahalanobis distance as described in Equation 3.11.

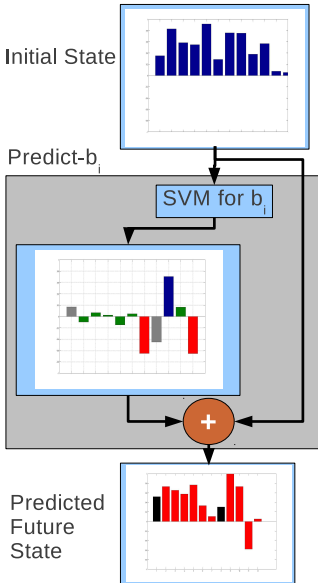


Figure 3.11: A sketch of how the future states are calculated. First, the current state of the object is fed to the trained SVM for the each behaviour. Then, the predicted effect's prototype is determined by applying the Equation 3.11 for each effect prototype. The mean value of the found effect is added on the initial features, with the exception of the inconsistent features, and the predicted future state is found. After this application, the predicted future state can be compared with other states; but the inconsistent features of the applied effect (denoted as black columns in predicted after-state) is excluded from the comparison calculations.

Planning toward achieving the goal is found using a breadth-first tree search. Starting with the initial state, we construct a tree such that it contains all the possible effect sequences with length n (n is empirically chosen as 3). The plan is made as the goal is matched with the predicted states after applying a sequence of behaviors, i.e. goal (see Figure 3.12 for a rudimentary example).

Given the object, we can obtain from the trained SVMs the behaviour that can achieve a desired effect with the highest probability (see Figure 3.11). Thus, we obtain the behaviours required for each step in the planned effect sequence, forming a sequence of behaviours. If

the obtained effect at any step in the behaviour sequence does not match with the expectation, then the planning restarts. The algorithm for multi-step planning is sketched in Algorithm 2.

Algorithm 2 Multi-step planning algorithm

```
e_current ← perceive()
B_sequence ← {}
b_winner ← −1
E_winner ← −1
level ← 1
f_current ← e_goal − e_current
{depthThreshold = 5 in our experiments} {d_thres is empirically determined}
while (level ≤ depthThreshold) ∧ (||f_current|| < d_thres) do
    d_min ← ∞
    for all E_j in ℰ do
        b_i ← svmbestbehavior(j, e_current)
        if modifiedMahalanobis then
            d_current ← d_mm(f_current, E_j)
        else if pureMahalanobis then
            d_current ← d_pm(f_current, E_j)
        else
            d_current ← d_e(f_current, E_j)
        end if
        if d_current < d_min then
            d_min ← d_current
            b_winner ← b_i
            E_winner ← E_j
        end if
    end for
    level ← level + 1
    B_sequence ← {B_sequence, b_winner}
    if modifiedMahalanobis then
        e_current ← e_current + f^{+,−,0}_{proto,E_winner}
    else
        e_current ← e_current + μ_{E_winner}
    end if
    f_current ← e_goal − e_current
end while
apply B_sequence on the object
report success or failure.
```

In cases where the robot can not reach to the object, this search method should report failure.

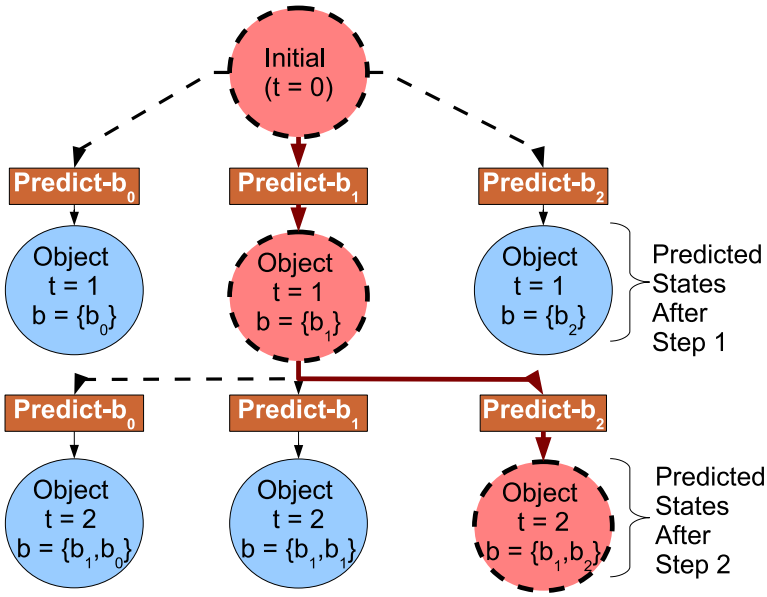


Figure 3.12: A simple depiction of how the planning is performed using the combinations of effects in the repertoire. At each step, the prediction block described in Figure 3.11 is applied for each behaviour. Once a future state close enough to goal state is obtained, the search is terminated.

For such practical purposes, we have empirically determined a fixed distance threshold to acknowledge successful planning, limited the search depth level to be 5, and provided novel effect instances which the robot can plan with at most 5 steps.

3.7 Methodology Summary

In this chapter, we have described the methodology we followed in this thesis. Equipped with six different behaviors and a 3D range sensor, iCub interacts with 35 different objects to collect effect information. The effect information is simply computed by taking the difference between the final and the initial states of the environment.

Using this information, we first proceed to learning verb, noun and adjective categories. To do so, we first train SVM classifiers for affordance learning. Then, using the findings, we proceed to form affordance vectors and train SVM classifiers to map affordance vector information to appropriate adjective and noun labels. Since SVM methodology is found to be not suitable for learning verb categories, we use Self-Organizing maps to learn verb categories.

We then proceed to conceptualize the learned categories. To do so, we perform unsupervised clustering on the mean and variance space of features from each category. With respect to the outcome of the clustering, we label the features as consistent or inconsistent and form prototype strings.

We evaluate our category learning via comparisons with simple learners and introducing novel objects and effects. For evaluating our conceptualization scheme, we first show the outcomes of novel effect predictions. Then, we perform multi-step planning, where a target state of the environment can be achieved through a proper sequence of behaviors. We compare the planning results with the methods that do not use prototype strings. The results and their evaluations are explained in Chapter 4, i.e., Experiments and Results.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, we evaluate our methods and present the results on noun, adjective and verb concepts. We then demonstrate how verb concepts can be used in multi-step planning.

4.1 Affordance Learning Results

In our experiments, the SVM classifiers for each behavior were trained with 5-fold cross validation, and were able to achieve accuracy values above of 90% for predicting the effect labels (i.e., what an object affords). As shown in Table 4.1, the variance values of k-fold peformance for each behavior-wise SVM classifier are within a reasonably small boundary. This suggests that our learning methodology offers a good generalizability. Having obtained such a result, we next discuss the results on category learning and concept representation results.

Table 4.1: Average, maximum and minimum prediction results for each behavior with respect to 5-fold cross validation.

Behavior	Average Rate	Maximum Rate	Minimum Rate
side-grasp	100%	100%	100%
top-grasp	90%	100%	85%
push-left	92%	100%	83%
push-right	96%	100%	85%
push-forward	100%	100%	100%
pull	96%	100%	86%

4.2 Verb Concepts

In this section, we evaluate our verb learning method.

4.2.1 Learning Verb Categories

The introduced language feature has played a great role in forming clusters. In comparison with the ground truth, the accuracies of *Type-0* SOM (without language), *Type-I* SOM, *Type-II* SOM, and *Type-III* SOM (which were described in Section 3.4.1) were 77.8 %, 90.5%, 99.9 % and 93.6%, respectively.

The prediction accuracy of the naïve learner remained at 55.69 % for both with and without reliefF preprocessing. Evidently, the number of instances and the distribution of features resulted in poor performance for multi-class learning with SVM. From these results, we infer that self-organizing map is a more suitable methodology in forming effect clusters for our work.

4.2.2 Representing Verb Categories

In our methodology, the prototype extraction on verbs is very sensitive to the distribution of the instances in each cluster. This is because as the number of misclassified instances increase, the mean and variance values for each feature, which are used in prototype extraction algorithm, will also change. As such, we proceeded with the results obtained with the predictions of *Type-II* SOM, which has the minimum number of misclassified instances. The extracted prototypes are shown in Table 4.2. In the Table, the effects are noted as MR: `Moved Right`, ML: `Moved Left`, MF: `Moved Forward`, P: `Pulled`, K: `Knocked`, NC: `No Change`, G: `Grasped`, D: `Disappeared`. The labels 0, -, +, * correspond to effect features with *Negligible Change*,*Consistent Decrease*, *Consistent Increase* and *Inconsistent Change*, respectively. The outcomes of these prototypes are as expected: the features that constitute the characteristics of the given effect label are clearly found as *consistently changed* (+ or -). Since there could be no features extracted from `disappeared` instances, the corresponding fields were padded with zero. As such, its prototype is filled with 0s (negligible changes) with the exception of object presence field (consistently decreased). In `No Change` instances, which are produced

where the object of interest was unreachable for the robot to perform complete action, the effect features are reported to be having either too small changes or inconsistent changes. In `Grasped` instances, since only the visual features were collected, the feature distribution resembled to the `No Change` cluster.

4.2.3 Interpreting Novel Interactions

Once the prototypes are obtained, it is straightforward to determine the novel effect instances (using Equation 3.11). Since this equation works on the change of features, it is independent of the presented object. Table 4.3 summarizes the distance information for the three example cases shown in Figure 4.1. We see that our effect distance Equation provides accurate predictions of effects, unlike non-prototype based evaluations with pure Mahalanobis and Euclidean distances. Furthermore, the time adds up in a pure Euclidean distance scheme as the number of effect instances in each cluster increase.

4.3 Adjective Concepts

In this section, we analyze the results obtained through our adjective learning methodology.

4.3.1 Learning Adjective Categories

An important point is whether adjectives should include explicit behaviour information (i.e., A_{48}-AL vs. A_8-AL). Theoretically, the performance of these models should converge while one-to-one, unique behavior-to-effect relations dominate the set of known affordances. In such cases, the behavior information would be redundant. On the other hand, with a behavior repertoire that may pose many-to-one-effect mappings, behavior information must be taken into account to obtain more distinguishable adjectives.

The comparison between the different adjective learning methods is displayed in Table 4.4, which displays the average 5-fold cross-validation accuracies. We see that the A_{48}-AL model performs better than A_8-AL and SAL. The reason that A_8-AL is worse than the other methods is eminent in Table 4.6, where we see that different adjective categories end up with similar descriptor vectors, losing distinctiveness. On the other hand, the A_{48}-AL model that has

Table 4.2: The effect prototypes. The effects are abbreviated as explained in Section 4.2.2

Effect Name	ΔAzimuth Histograms (20)	ΔZenith Histograms (20)	ΔCurvature Histograms	ΔShape Index Histograms	ΔPosition (x-y-z)	ΔOrient.	ΔSize (x-y-z)	ΔObject Presence
NC	*000000000 0000000000	0000000000 0000000*00	0000000000 00*0000000	0000000000 000000*000	000	0	000	0
MR	********** ******0*0*	********** *******000	0000000000 00********	*****0**** **0*0*0***	*+*	*	***	0
ML	******0000 00***0****	0**0**0000 **0000*000	0000000000 0*****0000	00*0**0*0* **0000*00*	0-0	*	000	0
MF	********** **********	******000* **000**000	0000000000 00****0***	***0*0**** **00000***	-00	*	000	0
P	********** **********	******000* **000**000	0000000000 00****0***	***0*0**** **00000***	+00	*	000	0
K	*0***00000 000000**0*	*000000000 0000000000	0000000000 000000000*	0000000000 0000000000	00-	*	00-	0
G	0000000000 0000000000	0000000000 0000000000	0000000000 0000000000	0000000000 0000000000	000	*	000	0
D	0000000000 0000000000	0000000000 0000000000	0000000000 0000000000	0000000000 0000000000	000	0	000	-

Table 4.3: Distances computed for effect instances in Figure 4.1 using the Equation 3.11 (first rows, abbreviated as d_{mm}), Equation 3.12 (second rows, abbreviated as d_e) and Equation 3.13 (third rows, abbreviated as d_{pm}). The label of the cluster with the smallest distance value is given as the prediction to the novel effect. The correct predictions are marked with bold green, whereas false predictions are marked with italic red. The effects are abbreviated as explained in Section 4.2.2

Figure	NC	MR	ML	MF	P	K	G	D
4.1(b) - d_{mm}	390.81	**146.24**	372.24	389.21	215.56	215.50	392.11	410.31
4.1(b) - d_e	237.01	236.89	237.42	237.24	237.42	237.42	237.25	*236.84*
4.1(b) - d_{pm}	392.16	**182.13**	386.92	416.43	241.06	219.28	395.04	410.31
4.1(d) - d_{mm}	731.36	494.18	416.42	**340.71**	393.76	358.06	738.04	790.41
4.1(d) - d_e	789.45	789.08	789.83	789.49	789.83	789.83	789.54	*788.84*
4.1(d) - d_{pm}	732.98	497.02	*417.18*	426.71	423.17	428.06	741.11	790.41
4.1(f) - d_{mm}	925.41	577.51	**267.45**	328.75	354.85	354.74	928.16	947.51
4.1(f) - d_e	946.74	946.42	947.03	946.66	947.03	947.03	947.01	*946.21*
4.1(f) - d_{pm}	929.37	580.26	**291.77**	369.75	373.37	359.88	929.94	947.51

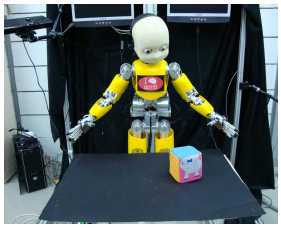
learned adjectives from the affordances of objects performs better than directly learning SAL model.

4.3.2 Representing Adjective Categories

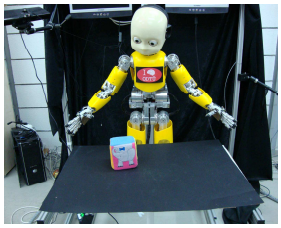
The Table 4.5 shows the prototypes of adjectives obtained when the model A_{48}-AL ($\mathcal{M}_a^1$) is used. The Table 4.6 shows the prototypes of adjectives obtained when the model A_8-AL ($\mathcal{M}_a^2$) is used.

Table 4.4: Average prediction results for the three adjective models in Sect. 3.4.

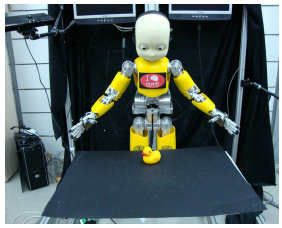
	A_{48}-AL $\mathcal{M}_a^1$	A_8-AL $\mathcal{M}_a^2$	SAL $\mathcal{M}_a^3$
Edgy-Round	87%	72%	89%
Short-Tall	93%	95%	89%
Thin-Thick	95%	72%	91%



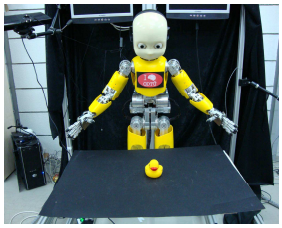
(a) Toy box initial state



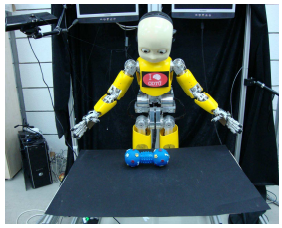
(b) Toy box final state



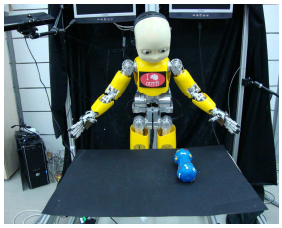
(c) Toy duck initial state



(d) Toy duck final state



(e) Toy bone initial state



(f) Toy bone final state

Figure 4.1: Some novel instances for effects. The effect is simply the difference of final and initial states of the given object. The novel effect predictions with different distance metrics for these instances are conveyed on Table 4.3

These prototypes allow iCub to relate adjectives with what it can and cannot do with them. We see from the Table what behaviours can consistently generate which effects on which types of objects (specified with their adjectives). For example, with a consistently large probability, the robot would generate *no change* effect on *edgy* or *thick* objects when *top grasp* behavior was applied. Furthermore, the *short* and *tall* objects show a clear distinction in response to pushing behaviors (*tall* objects have a high probability to be *knocked* while *short* objects simply get pushed).

The comparison between the different adjective learning methods is displayed in Table 4.4, which displays the average 5-fold cross-validation accuracies. We see that the A_{48}-AL model performs better than A_8-AL and SAL. The reason that A_8-AL is worse than the other methods is eminent in Table 4.6, where we see that different adjective categories end up with similar descriptor vectors, losing distinctiveness. On the other hand, the A_{48}-AL model that has

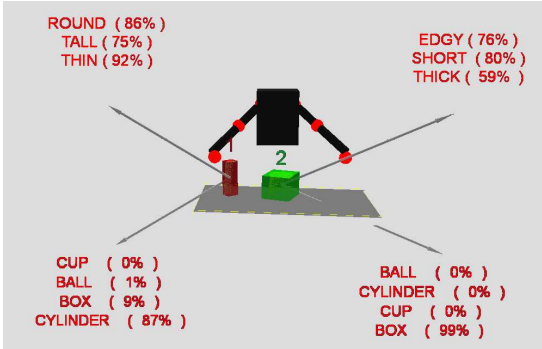


Figure 4.2: After learning nouns and adjectives, iCub can refer to an object with its higher level representations or understand what is meant if such representations are used by a human.

Table 4.5: The prototypes of adjectives for the model A_{48}-AL ($\mathcal{M}_a^1$). TG: Top Grasp, SG: Side Grasp, PR: Push Right, PL: Push Left, PF: Push Forward, PB: Pull. For each behavior, there are eight effect categories: *a*: Moved Right, *b*: Moved Left, *c*: Moved Forward, *d*: Pulled, *e*: Knocked, *f*: No Change *g*: Grasped, *h*: Disappeared.

Adjective	TG *abcdefgh*	SG *abcdefgh*	PR *abcdefgh*	PL *abcdefgh*	PF *abcdefgh*	PB *abcdefgh*
Edgy	-----+--	-----**-	*---**-+	-*--**-+	---***-+	---*++-+
Round	-----**-	-----+--	*---+*-+	-*--+*-+	---**+-*	---**+-*
Short	-----**-	-----+--	+---**-+	-+--**-+	---+**-+	---+*+-+
Tall	-----**-	-----**-	*---+*-+	-*--+*-+	---*++-*	---*++-*
Thin	-----**-	-----**-	*---+*-+	-*--+*-+	---*+*-+	----++-+
Thick	-----+--	-----**-	*---**-*	-*--**-*	---**+-*	---+*+-*

Table 4.6: The prototypes of adjectives for the model A_8-AL ($\mathcal{M}_a^2$). The effects are abbreviated as explained in Section 4.2.2

Adjective	MR	ML	MF	P	K	NC	G	D
Edgy	$*$	$*$	$*$	$*$	$+$	$+$	$*$	$*$
Round	$*$	$*$	$*$	$*$	$*$	$+$	$+$	$+$
Short	$*$	$*$	$*$	$*$	$*$	$+$	$+$	$+$
Tall	$-$	$-$	$-$	$-$	$+$	$+$	$+$	$+$
Thin	$*$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
Thick	$*$	$*$	$*$	$*$	$*$	$*$	$+$	$+$

learned adjectives from the affordances of objects performs better than directly learning SAL model.

4.3.3 Adjectives of Novel Entities

Table 4.7 shows the predicted adjectives from the different models on novel objects. We see that, for adjectives, $\mathcal{M}_a^1$ is better in naming adjectives than $\mathcal{M}_a^2$. For example, $\mathcal{M}_a^2$ misclassifies object-5 as *edgy*, object-7 as *thin* and object-1 as *thick* whereas $\mathcal{M}_a^1$ correctly names them. On some objects (e.g., object-3), where there are disagreements between the models, correctness cannot be evaluated due to the complexity of the object. If we look at the direct mapping from objects' appearance to adjectives ($\mathcal{M}_a^3$), we see that it misclassifies object-7 as *round*, object-6 as *tall* and objects 2 and 8 as *edgy*. Compared to the simple learner, our model can predict the adjectives of novel objects with a higher precision.

4.4 Noun Concepts

In this section, we analyze the results obtained through our noun learning methodology.

4.4.1 Learning Noun Categories

For the three models trained on nouns (Sect. 3.4.3), we get the following 5-fold cross-validation accuracies: A_{48}-NL: 87.5%, A_8-NL: 78.1% and SNL: 94%. We see that, unlike the case in adjectives, directly learning the mapping from appearance to nouns performs better than using the affordances of objects. This suggests that the affordances of the objects (used

Table 4.7: Predicted adjectives for novel objects using 3 different models (bold labels denote correct classifications).

ID	Object	A_{48}-AL $\mathcal{M}_a^1$	A_8-AL $\mathcal{M}_a^2$	SAL $\mathcal{M}_a^3$
1		**edgy (54 %)** **short (97 %)** **thin (59 %)**	**edgy (89 %)** **short (91 %)** **thick (52 %)**	**edgy (89 %)** **short (55 %)** **thin (52 %)**
2		**round (77 %)** **short (77 %)** **thin (89 %)**	**round (90 %)** **short (91 %)** **thin (67 %)**	edgy (79 %) **short (42 %)** **thin 67 %**
3		**edgy (63 %)** short (94 %) **thin (96 %)**	round (72 %) short (92 %) **thin (72 %)**	**edgy (64 %)** **tall (67 %)** **thin 84 %**
4		**round (84 %)** **short (98 %)** **thick (91 %)**	edgy (94 %) **short (% 87)** thin (68 %)	**round (77 %)** **short (68%)** thin (62 %)
5		**round (84 %)** **short (97 %)** **thick (95 %)**	edgy (81 %) **short (93 %)** **thick (59 %)**	**round (89 %)** **short (67 %)** **thick (58 %)**
6		**edgy (84 %)** **short (98 %)** **thin (92 %)**	**edgy (79 %)** **short (80 %)** **thin (79 %)**	**edgy (79 %)** tall (45 %) thick (62 %)
7		**edgy (62 %)** **short (98 %)** **thick (78 %)**	**edgy (52 %)** **short (93 %)** thin (53 %)	round (84 %) **short (54 %)** **thick (68 %)**
8		**round (72 %)** **short (98 %)** **thick (79 %)**	**round (69 %)** **short (95 %)** **thick (64 %)**	edgy (89 %) **short (67 %)** **thick (52 %)**

in our experiments) are less descriptive for the noun labels we have used.

4.4.2 Representing Noun Categories

Although our functional noun learner performs worse in terms of prediction, the extracted prototypes provide a fair distinction between different noun labels. Similar with the case in verbs, we have chosen to pick the learning model with the higher prediction rate to provide the noun clusters (i.e., A_{48}-NL). Table 4.8 shows the prototypes derived for nouns *cup*, *box*, *cylinder* and *ball*.

4.4.3 Nouns of Novel Entities

Table 4.9 shows the results obtained on novel objects. Unlike the case in adjectives, the simple learner (SNL) significantly outperforms the A_{48}-NL and A_8-NL models. Hence, we conclude that the set of nouns (cup, cylinder, box, ball) we have are more of appearance-based.

4.5 Multi-Step Planning

In this section, we demonstrate (i) that verb concepts are very useful in making multi-step plans for reaching to a given target state, and (ii) that prototype-based representation for concepts is better than exemplar-based concept representation (see 2.2.1 for a review).

The multi-step planning results are provided in Figure 4.3. We see that, using the verb concepts presented in Table 4.2, iCub can successfully find a sequence of effect prototypes leading to the target state. From these prototypes, iCub can choose the best behaviors yielding those effect prototypes.

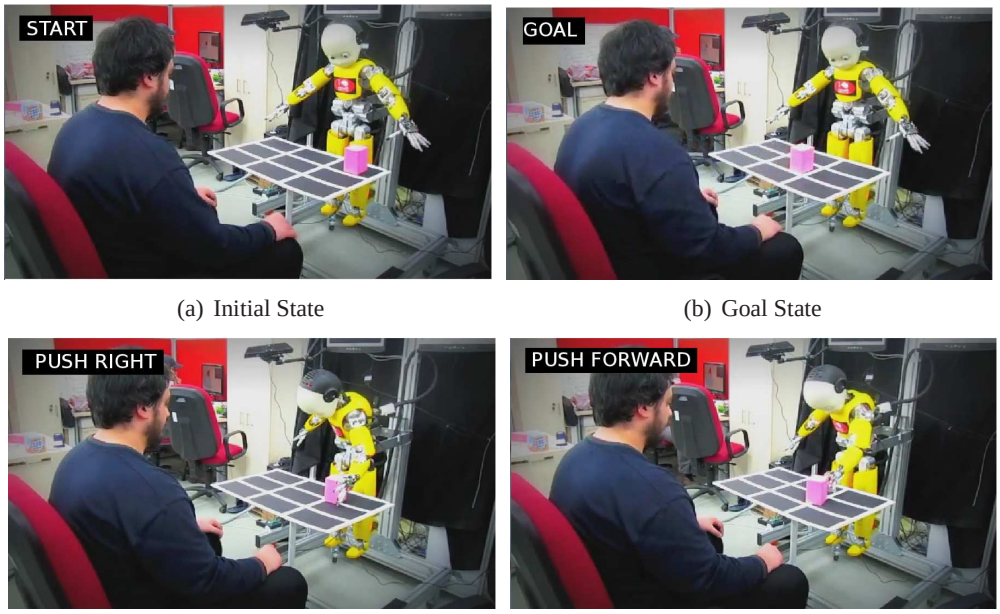
In Figures 4.5 and 4.6, we provide the planning results when an exemplar-based representation is used for concepts. We see that planner could not produce proper sequences of behavior to achieve the given goals (the distance threshold was constant throughout the experiments) As such, planner could not produce proper sequences of behavior to achieve the given goals (the distance threshold was constant throughout the experiments).

Table 4.8: The noun strings. The labels 0, + and * correspond to *Consistently Small*, *Consistently Large* and *Inconsistent* features.

Noun	Azimuth Histograms (20)	Zenith Histograms (20)	Curvature Histograms	Shape Index Histograms	Position (x-y-z)	Orient.	Size (x-y-z)	Object Presence
Cup	0000000000 0000000000	00000+0000 0000000000	0++0+**000 0000000000	0000000000 ++00000000	000	0	000	+
Box	000+++000+ 0000000000	+000000000 +0000+0000	00000**000 0000000000	00+0000000 0+0+++0000	000	0	000	+
Cyl.	0+0+000000 0000000000	0000000000 0000000000	00000**000 0000000000	0000000000 ++0+00+000	000	0	000	+
Ball	0000000000 00+0000000	000000+000 0+++000000	00000**000 0000000000	000++00000 ++00000000	000	0	000	+

Table 4.9: Noun prediction for novel objects using 3 different models. Bold labels denote the correct classifications with respect to the ground truth (see Table 4.7 for pictures of the objects).

ID	A_{48}-NL	A_8-NL	SNL
1	**box (74 %)**	cylinder (42 %)	**box (97 %)**
2	**ball (83 %)**	**ball (44 %)**	**ball (97 %)**
3	**cylinder (87 %)**	**cylinder (39 %)**	**cylinder (95 %)**
4	box (94 %)	**cylinder (38 %)**	**cylinder (86 %)**
5	**box (89 %)**	cylinder (35 %)	**box (94 %)**
6	cup (89 %)	**cylinder (44 %)**	box (46 %)
7	**box (89 %)**	**box (32 %)**	**box (93 %)**
8	**cup (89 %)**	cylinder (44 %)	**cup (98 %)**



(a) Initial State (b) Goal State

(c) First Step (d) Second and Final Step

Figure 4.3: A sample execution of a multi step planning. First, the start position (4.3(a)) and final position is shown (4.3(b)). The robot plans out and applies push right (4.3(c)) and push forward (4.3(d) When a simple Euclidian distance is used, such a sequence of behaviors could not be planned).

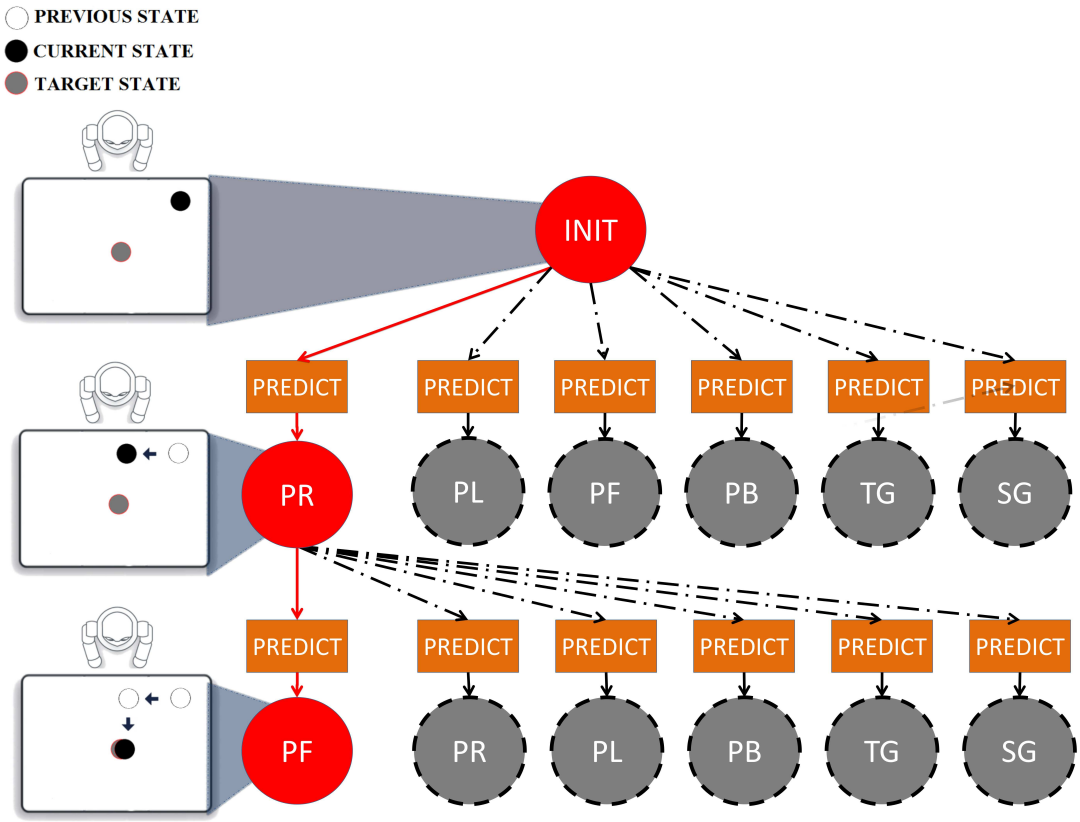


Figure 4.4: A multi-step planning trial with the modified Mahalanobis distance metric. The behaviors are abbreviated as *PR* (*push-right*), *PL* (*push-left*), *PF* (*push-forward*), *PB* (*pull*), *TG* (*top-grasp*), *SG* (*side-grasp*). The planner successfully terminates with a reasonably small sequence of behaviors executed. The trial is also visualized in Figure 4.3

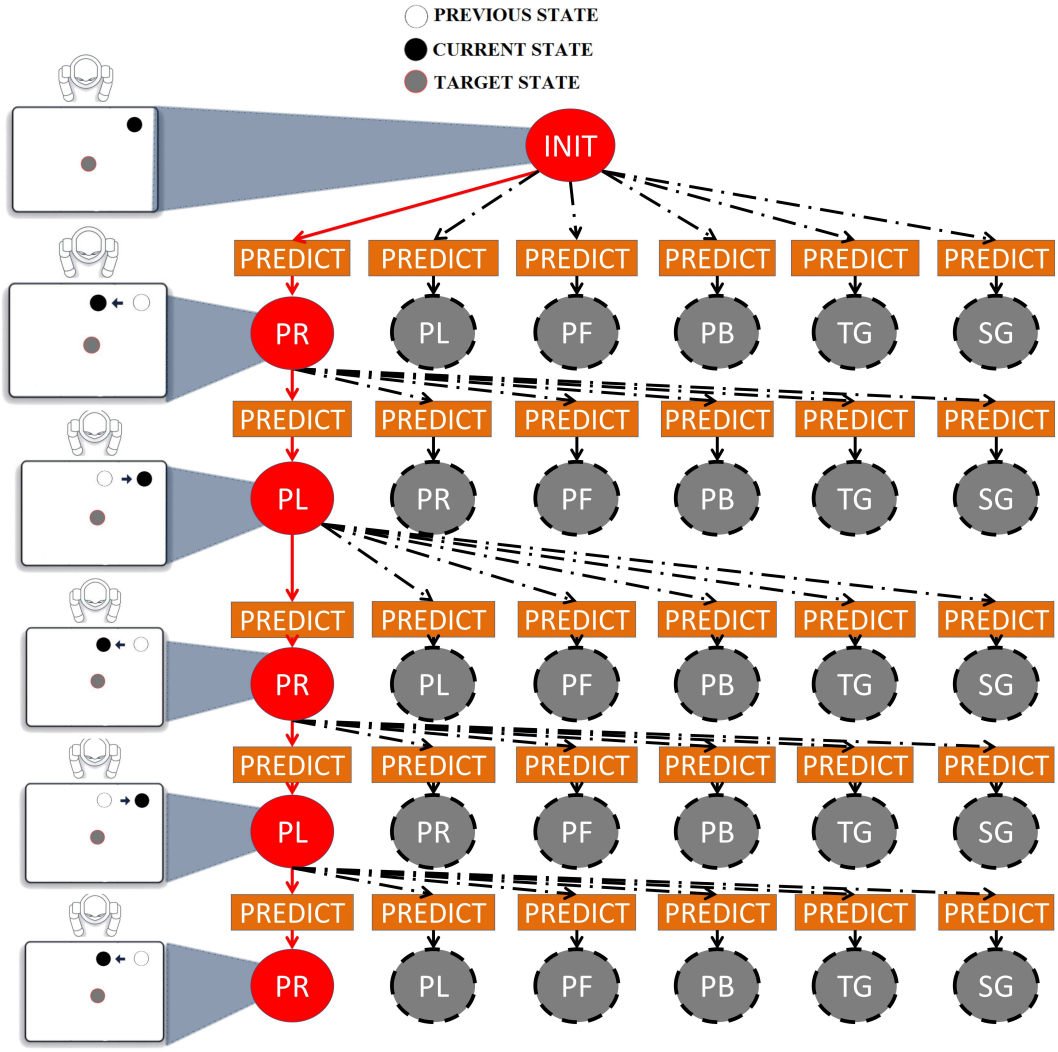


Figure 4.5: Multi-Step planning trial with pure mahalanobis distance. The behaviors are abbreviated as *PR* (*push-right*), *PL* (*push-left*), *PF* (*push-forward*), *PB* (*pull*), *TG* (*top-grasp*), *SG* (*side-grasp*). The initial and the target states for the objects are the same with the one provided in planning in Figure 4.3. Since the distance calculations yield wrong results, the search does not terminate with success.

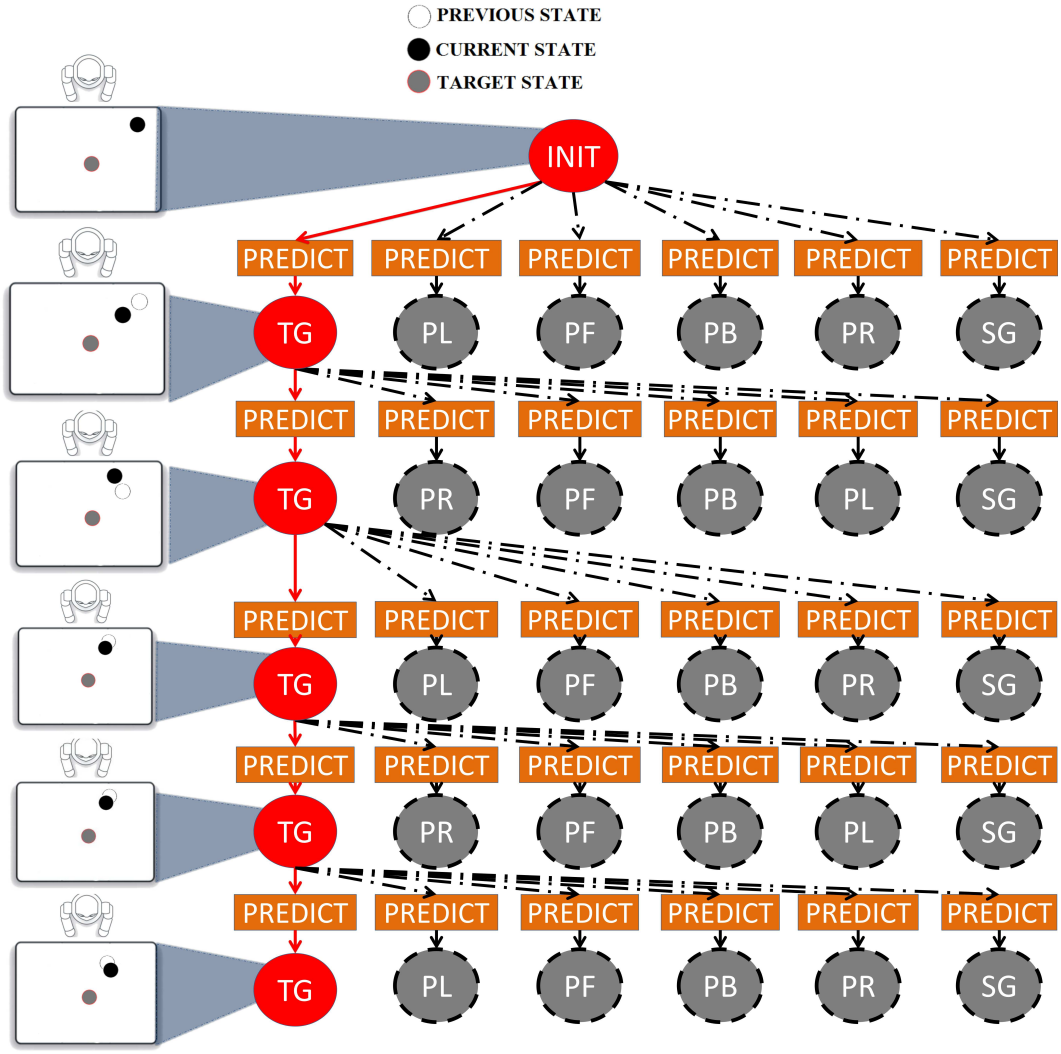


Figure 4.6: Multi-Step planning trial with pure euclidean distance. The behaviors are abbreviated as *PR* (*push-right*), *PL* (*push-left*), *PF* (*push-forward*), *PB* (*pull*), *TG* (*top-grasp*), *SG* (*side-grasp*). The initial and the target states for the objects are the same with the one provided in planning in Figure 4.3. Since the distance calculations yield wrong results, the search does not terminate with success.

4.6 Computational Complexity

The critical analysis results in terms of computational complexity are obtained with the choice of distance metric for multi-step planning. Even with a simple theoretical analysis, it is apparent that using Equation 3.11 or 3.13 makes the system asymptotically faster than with using Equation 3.12. This is since Equations 3.11 and 3.13 only use the mean values of the categories. The run-time complexities for executions of Equations 3.11 and 3.13 are $\Theta(N \times |\mathcal{E}|)$, where $\mathcal{E}$ is the set of all observed effects and N is the number of features extracted from the interactions. The discarding of inconsistent features makes the execution of Equation 3.11 faster than of Equation 3.13 with a constant factor. However, if no features are discarded with respect to the prototypes, then the run-time complexities of using these equations are exactly the same (this case did not occur within our experimental results). In contrast to the execution of these two equations, the run-time complexity for Equation 3.12 is $\Theta(N \times \sum_{E_j \epsilon \mathcal{E}} |E_j|)$, where $|E_j|$ denotes the number of instances in the effect category E_j. Thus, the usage of pure Euclidean distance significantly deteriorates the performance of the system as the number of instances increase. The discussion for computational complexity is finalized in Chapter 5.

CHAPTER 5

DISCUSSION AND CONCLUSION

In this chapter, we provide an overall discussion about the outcomes and limitations of the system, and sketch some possible future extensions.

In this work, we have shown how a humanoid robot can learn concepts of a subset of verbs, nouns and adjectives through its interactions with the environment. We have followed the affordance formalization framework of Sahin et al. [11], which represents affordances as a triplet of entity, behavior and effect. We extended the previously used model to learn adjective and noun (i.e., object) concepts in terms of affordances. Our observations show that a subset of nouns and adjectives can be inferred from and plausibly represented in terms of affordances.

Following the previous studies in KOVAN research Laboratory, we represented these concepts as strings, which clearly show the distinctions of different concepts. Using prototypes, we have been successful in eliminating irrelevant features for our conceptual representations. Thus, our system has provided us some insight to redeem problems in dealing with high-dimensional data. On the other hand, since our experiments are already limited even within the domain of robotics, we do not claim having offered a generic and conclusive methodology to overcome the Curse of Dimensionality. There is always a room for improvement in this aspect of our computational system.

Furthermore, we have used our effect prototypes to prove the superiority of Prototype-Based view of concepts over Exemplar-Based view of concepts. We demonstrated this superiority by first examining novel interactions and then designing an accurate forward chain planner for goal emulation that uses effect prototypes.

We have proposed and tested two main adjective learning approaches, one of which uses

explicit behavior information and the other one does not. Our experiments evaluated the different representations (a more discriminative and a more generic, respectively) and performances obtained with these approaches. The usage of affordance estimates and explicit behavior information has shown a notable increase in representational power and accuracy.

While we have successfully obtained proper representations for adjectives and nouns, it is important to recall that these representations are subject to the sensorimotor limitations of the robot. These limitations are maintained by the number, type and the quality of the behaviors in the action repertoire and the properties of the perceptual subsystem. As such, there is always a room for improvement in functional object categorization. For example, had we included a behavior to try to fill objects with some liquid, the *cups* concept would be much easier to be formed and predicted. Thus, we see that, if any kind of supervision is to be applied in our system, it must be closely related to the sensorimotor capabilities of the robot. In a similar manner, we have presented a limited number of adjectives for practical purposes. This set of adjectives can be considerably extended as the behavior repertoire is enriched and hence the new effects are observed. It should be noted once again that, however, we only tested our system with a *subset* of possible behaviors. We do not claim and prove that all behaviors are applicable for our system. Further verifications and, if possible, extensions might be required to arrange our system for all kinds of behaviors. The methodology for such verifications is itself an important problem.

The language feature has been a very relevant feature for effects, hence its inclusion yielded clusters with close correspondance to ground truth. It is also reasonable to assert that this inclusion reduced the level of supervision in the system to some extent. While this is also verified with various subsets of our features, this approach can be questioned for whether it can be a permanent solution. After all, the dominance of this feature decreases while the number of features increase.

Another issue to consider is the generalizability of our methodologies. By reporting results on novel data, we have implicitly shown that our category learning methods are feasible for generalizing. The proof for generalizability can be improved by providing extensive theoretical analysis. Such analysis can be designed through, for example, examining variations in cross validation accuracy results, which is reported in Section 4.1. The generalizability for our methodology can be also tested by extending the system with additional tasks that utilize

affordances, e.g., tool use [70, 80, 81] or even learning primitive motor functions [69, 82, 83]. In related literature, such tasks are implemented and reported to be feasible for developmental robotics.

In terms of time complexity, the bottleneck for our system lies in the data collection procedure. This is due to various constaints such as safety measures, simplicity of behaviors, and the need for producing repetitive data. Application of Self-Organizing Maps and Robust Growing Neural Gas algorithms are the secondary bottlenecks. This is because they are stochastic algorithms (hence multiple runs were taken) and they involved repetitive feeding of data. The remaining tasks, such as category learning for nouns and adjectives and multi step planning were relatively cheap to execute. For example, with the advent of the built-in libraries for SVM [96], the category learning for adjectives and nouns could be handled within a negligible amount of time. One advantage for our system appears in distance calculations: due to the nature of distance equations mentioned in Chapter 3, our system performs asymptotically faster when the prototype-based view is adopted.

5.1 Future Work

In this thesis, we have obtained the effect information by simply taking differences between final and initial states of objects. A critical improvement would be processing the information *during* the application of behaviors to obtain affordances in a continuous time scale. The analysis of affordances in this manner may give way to obtain *adverb* concepts, where we can find further distinctions between features that change differently over time, e.g. *slowly* versus *fast*, *uniformly* versus *with fluctuations*.

Another immediate improvement would be towards determining further practical applications for concepts. The odd-one-out task, for example, is a quite suitable application to utilize noun and adjective concepts. In this task, the robot would be presented a number of objects and asked to find out the object that does not belong to the group formed by the rest. Using the string representations, any number of given objects can be compared and grouped in an ad-hoc manner. The larger the objects' concept repertoire, the more likely that the system can find the most odd object in a given scene. A further usage of these concepts would be relating objects in a functional sequence. Working on extracting the semantics of sentences such as

"Push the edgy box with thin cylinder", our computational model may lead to an alternative approach of *tool affordances*.

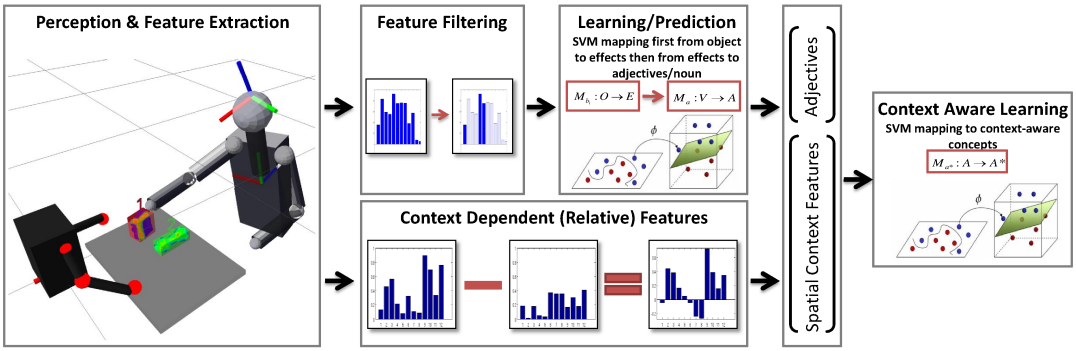


Figure 5.1: The proposed context-aware adjective learning system. iCub learns adjectives based on the affordance predictions and contextual features.

The affordance formalization framework also appears to be useful in constructing a context-aware adjective learning system. Given data consisting of pairs of objects available in a scene, we developed a learning model that trains on the affordances and spatial differences of objects (see Figure 5.1). In this manner, the robot has successfully learned adjective tags such as *relatively* tall or short, *relatively* round or edgy and *relatively* thin or thick. Furthermore, this model has performed surprisingly better than a baseline learner (that only trains on perceptual differences of the object pairs) on novel instances. As the ground truth for the evaluations are derived from multiple human attendants, this kind of extension to our work may establish a more customizable learning system. As such, we can extend our computational system towards addressing issues in human-robot interaction problems.

A different direction could be towards the improvement of obtaining and processing of features. Currently, the feature set consists of a mixture of purely extracted (e.g. position and size) and preprocessed features (e.g. language feature). For a greater challenge, the system could be designed such that all features are derived from purely low-level information. Then, these features can be used to form second-order feature vectors that have more abstract semantics. The architectures reviewed in Chapter 2 such implementations are possible, and are plausible in the perspective of cognitive systems. In a similar sense, the proposed architecture may be tested for whether it can produce meaningful results when the feature set is divided according to sensor modalities, such as *auditory*, *tactile*, *colour*, *shape*, *pose*. Existing approaches hint that this methodology can also be quite useful in forming concepts.

REFERENCES

[1] P. Haazebroek, S. van Dantzig, and B. Hommel, "A computational model of perception and action for cognitive robotics," *Cognitive Processing*, vol. 12, 2011.

[2] A. M. Glenberg and V. Gallese, "Action-based language: A theory of language acquisition, comprehension, and production," *Cortex*, 2011.

[3] E. Ugur, E. Oztop, and E. Şahin, "Goal emulation and planning in perceptual space using learned affordances," *Robotics and Autonomous Systems*, vol. 59, no. 7-8, 2011.

[4] A. F. Morse, P. Baxter, T. Belpaeme, L. B. Smith, and A. Cangelosi, "The power of words," in *Proceedings of Epigenetic Robotics*, 2011.

[5] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335 – 346, 1990.

[6] ROSSI, "Emergence of communication in robots through sensorimotor and social interaction, eu fp7-ict 216125," 2008-2011.

[7] I. Atil, "Function and appearance-based emergence of object concepts through affordances. a thesis submitted to graduate school of natural and applied sciences," 2010.

[8] N. Dag, "Emergence of verb and object concepts through learning affordances. a thesis submitted to graduate school of natural and applied sciences," 2010.

[9] J. Gibson, *The ecological approach to visual perception*. Lawrence Erlbaum, 1986.

[10] E. J. Gibson, "Perceptual learning in development: Some basic concepts," *Ecological Psychology*, vol. 12, no. 4, pp. 295–302, 2000.

[11] E. Sahin, M. Cakmak, M. Dogar, E. Ugur, and G. Ucoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, pp. 447–472, 2007.

[12] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: A survey," in *IEEE Transactions on Autonomous Mental Development*, vol. 1, pp. 12–34, IEEE, 2009.

[13] A. Stoytchev, "Some basic principles of developmental robotics," in *IEEE Transactions on Autonomous Mental Development*, vol. 1, pp. 122–130, IEEE, 2009.

[14] V. Sukhoy, J. Sinapov, L. Wu, and A. Stoytchev, "Learning to press doorbell buttons," in *Proceedings of International Conference on Developmental Learning*, pp. 132–139, 2010.

[15] A. S. J Sinapov, "The odd one out task: Toward an intelligence test for robots," in *Cognitive Science*, pp. 126–131, 2010.

[16] "Autoincsfa and vision-based developmental learning for humanoid robots," in *V R Kompella, Leo Pape, J Masci, M Frank, Jürgen Schmidhuber*, 2011.

[17] J. R. Simon and J. D. Wolf 1963.

[18] M. K. Masahiko Haruno, Daniel M Wolpert, "Hierarchical mosaic for movement generation," *International Congress Series*, vol. 1250, pp. 575–590, 2003.

[19] M. K. DM Wolpert, K Doya, "A unifying computational framework for motor control and social interaction," in *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences*, vol. 358, pp. 593–602, The Royal Society, 2003.

[20] L. Gabora, E. Rosch, and D. Aerts, "Toward an ecological theory of concepts," *Ecological Psychology*, vol. 20, no. 1, pp. 84–116, 2008.

[21] J. Kruschke, "Category learning," *The handbook of cognition, Eds: K. Lamberts, R. L. Goldstone*, pp. 183–201, 2005.

[22] E. H. Rosch, "Natural categories," *Cognitive psychology*, vol. 4, no. 3, pp. 328–350, 1973.

[23] J. Rouder and R. Ratcliff, "Comparing exemplar-and rule-based theories of categorization," *Current Directions in Psychological Science*, vol. 15, no. 1, pp. 9–13, 2006.

[24] J. Bruner, J. Goodnow, and G. Austin, *A study of thinking*. Transaction Publishers, 1986.

[25] F. G. Ashby and W. T. Maddox, “Relations between prototype, exemplar, and decision bound models of categorization,” *Journal of Mathematical Psychology*, vol. 37, no. 3, pp. 372–400, 1993.

[26] R. Nosofsky, J. Kruschke, and S. McKinley, “Combining exemplar-based category representations and connectionist learning rules,” *Learning, Memory*, vol. 18, no. 2, pp. 211–233, 1992.

[27] J. Minda and J. Smith, “Prototypes in category learning: The effects of category size, category structure, and stimulus complexity,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27, no. 3, pp. 775–799, 2001.

[28] R. Nosofsky and S. Zaki, “Exemplar and prototype models revisited: Response strategies, selective attention, and stimulus generalization,” *Learning, Memory*, vol. 28, no. 5, pp. 924–940, 2002.

[29] D. Leopold, A. O’Toole, T. Vetter, and V. Blanz, “Prototype-referenced shape encoding revealed by high-level aftereffects,” *Nature Neuroscience*, vol. 4, no. 1, pp. 89–94, 2001.

[30] M. Johansen and J. Kruschke, “Category representation for classification and feature inference,” *Learning, Memory*, vol. 31, no. 6, pp. 1433–1458, 2005.

[31] Y. Rosseel, “Mixture Models of Categorization,” *Journal of Mathematical Psychology*, vol. 46, no. 2, pp. 178–210, 2002.

[32] J. Parthemore and A. Morse, “Representations reclaimed: Accounting for the co-emergence of concepts and experience,” *Pragmatics & Cognition*, vol. 18, no. 2, pp. 273–312, 2010.

[33] P. Gärdenfors, *Conceptual spaces: The geometry of thought*. The MIT Press, 2004.

[34] S. Harnad, “The symbol grounding problem,” *Physica*, vol. D, no. 42, pp. 335–346, 1990.

[35] A. M. Borghi, “Object concepts and embodiment: Why sensorimotor and cognitive processes cannot be separated,” *La nuova critica*, vol. 15, no. 4, pp. 447–472, 2007.

[36] A. Cangelosi and S. Harnad, “The adaptive advantage of symbolic theft over sensorimotor toil: Grounding language in perceptual categories,” *Evolution of Communication*, vol. 4, no. 1, pp. 117–142, 2001.

[37] M. Fischer and R. Zwaan, “Embodied language: A review of the role of the motor system in language comprehension,” *The Quarterly Journal of Experimental Psychology*, vol. 61, no. 6, pp. 825–850, 2008.

[38] V. Gallese and G. Lakoff, “The brain’s concepts: The role of the sensory-motor system in conceptual knowledge,” *Cognitive neuropsychology*, vol. 22, no. 3, pp. 455–479, 2005.

[39] A. Cangelosi and T. Riga, “An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots,” *Cognitive science*, vol. 30, no. 4, pp. 673–689, 2006.

[40] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, L. Fadiga, B. Wrede, K. Rohlfing, E. Tuci, K. Dautenhahn, J. Saunders, and A. Zeschel, “Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics,” *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 167–195, 2010.

[41] L. Steels, “Evolving grounded communication for robots,” *Trends in Cognitive Science*, vol. 7, no. 7, pp. 308–312, 2003.

[42] A. Glenberg and M. Kaschak, “Grounding language in action,” *Psychonomic Bulletin & Review*, vol. 9, no. 3, p. 558, 2002.

[43] A. Cangelosi, “Grounding language in action and perception: From cognitive agents to humanoid robots,” *Physics of Life Reviews*, vol. 7, no. 2, pp. 139–151, 2010.

[44] A. Glenberg, M. Sato, L. Cattaneo, L. Riggio, D. Palumbo, and G. Buccino, “Processing abstract language modulates motor system activity,” *The Quarterly Journal of Experimental Psychology*, vol. 61, no. 6, pp. 905–919, 2008.

[45] R. Zwaan and L. Taylor, “Seeing, acting, understanding: Motor resonance in language comprehension,” *Journal of Experimental Psychology-General*, vol. 135, no. 1, pp. 1–11, 2006.

[46] G. Rizzolatti and M. Arbib, "Language within our grasp," *Trends in neurosciences*, vol. 21, no. 5, pp. 188–194, 1998.

[47] "Collective consciousness and the social brain," *Journal of Consciousness Studies*, vol. 15, pp. 264–276, 2008.

[48] V. Gallese and A. Goldman, "Mirror neurons and the simulation theory of mind-reading," *Trends in cognitive sciences*, vol. 12, pp. 493–501, 1998.

[49] G. Rizzolatti and L. Craighero, "The mirror-neuron system," *Annual Review of Neuroscience*, vol. 27, no. 1, pp. 169–192, 2004.

[50] C. Yu and D. H. Ballard, "On the integration of grounding language and learning objects," in *Proc. 19th Int. Conf. on Artifical Intelligence*, AAAI'04, pp. 488–493, AAAI Press, 2004.

[51] P. Carbonetto and N. de Freitas, "Why can't jose read? the problem of learning semantic associations in a robot environment," in *Proc. the HLT-NAACL 2003 workshop on Learning word meaning from non-linguistic data*, pp. 54–61, 2003.

[52] C. L. N. J. Saunders and C. Lyon, "Robot learning of lexical semantics from sensorimotor interaction and the unrestricted speech of human tutors," in *Proc. 2nd Int. Symp. New Frontiers HumanRobot Interact. ASIB Convent.*, 2010.

[53] A. Cangelosi, "Evolution of communication and language using signals, symbols, and words," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 2, pp. 93–101, 2001.

[54] S. Nolfi and D. Marocco, "Active perception: A sensorimotor account of object categorization," in *From Animals to Animats 7: Proceeding on the Sixth International Conference on Simulation of Adaptive Behavior*, 2002.

[55] S. Jivko, B. Taylor, S. Connor, O. Ugonna, G. Shane, and S. Alexander, "Interactive object recognition using proprioceptive and auditory feedback," *The International Journal of Robotics Research*, vol. Published Online, 2011.

[56] N. Dağ, I. Atıl, S. Kalkan, and E. Şahin, "Learning affordances for categorizing objects and their properties," *IEEE International Conference on Pattern Recognition (ICPR)*, 2010.

[57] P. Cohen, C. Morrison, and E. Cannon, "Maps for verbs: The relation between interaction dynamics and verb use," in *Proceedings of the Nineteenth International Conference on Artificial Intelligence (IJCAI 2005)*, 2005.

[58] V. Krunic, G. Salvi, A. Bernardino, L. Montesano, and J. Santos-Victor, "Affordance based word-to-meaning association," *IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 4138–4143, 2009.

[59] D. Marocco, A. Cangelosi, K. Fischer, and T. Belpaeme, "Grounding action words in the sensorimotor interaction with the world: experiments with a simulated icub humanoid robot," *Frontiers in Neurorobotics*, vol. 4, no. 7, pp. 1–15, 2010.

[60] M. Rudolph, M. Muhlig, M. Gienger, and H.-J. Bohme, "Learning the consequences of actions: Representing effects as feature changes," *Int. Symposium on Learning and Adaptive Behavior in Robotic Systems*, 2010.

[61] H. Kozima, C. Nakagawa, and H. Yano, "Emergence of imitation mediated by objects," *Lund University Cognitive Studies*, pp. 59–61, 2002.

[62] C. L. Nehaniv, C. Lyon, and A. Cangelosi, "Current work and open problems: A roadmap for research into the emergence of communication and language," in *Emergence of Communication and Language* (C. L. N. C. Lyon and A. Cangelosi, eds.), pp. 1–27, Springer, 2007.

[63] M. Christiansen and S. Kirby, "Language evolution: Consensus and controversies," *Trends in Cognitive Sciences*, vol. 7, no. 7, pp. 300–307, 2003.

[64] C. Lyon, C. Nehaniv, and A. Cangelosi, *Emergence of communication and language*. Springer-Verlag New York Inc, 2007.

[65] A. Cangelosi, E. Hourdakis, and V. Tikhanoff, "Language acquisition and symbol grounding transfer with neural networks and cognitive robots," in *IJCNN'06. International Joint Conference on Neural Networks*, pp. 1576–1582, 2006.

[66] A. Cangelosi and D. Parisi, "The processing of verbs and nouns in neural networks: Insights from synthetic brain imaging," *Brain and Language*, vol. 89, no. 2, pp. 401–408, 2004.

[67] T. Hashimoto and A. Masumi, “Learning and transition of symbols: Towards a dynamical model of a symbolic individual,” in *Emergence of Communication and Language* (C. N. C. Lyon and A. Cangelos, eds.), pp. 223–236, Springer, 2007.

[68] L. Steels, “The recruitment theory of language origins,” in *Emergence of Communication and Language* (C. L. N. C. Lyon and A. Cangelosi, eds.), pp. 129–150, Springer, 2007.

[69] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, “Learning object affordances: From sensory–motor coordination to imitation,” *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 15–26, 2008.

[70] A. Stoytchev, “Learning the affordances of tools using a behavior-grounded approach,” *Towards Affordance-Based Robot Control*, pp. 140–158, 2008.

[71] D. Kraft, N. Pugeault, E. Baseski, M. Popovic, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger, “Birth of the object: Detection of objectness and extraction of object shape through object action complexes,” *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 247–265, 2008.

[72] E. Ugur and E. Şahin, “Traversability: A case study for learning and perceiving affordances in robots,” *Adaptive Behavior*, vol. 18, no. 3-4, pp. 258–284, 2010.

[73] E. S. Asil Kaan Bozcuoglu, “Traversability on a simple humanoid : What did i just trip over?,” in *11th IEEERAS International Conference on Humanoid Robots*, pp. 701–706, IEEE, 2011.

[74] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, “Learning about objects through action - initial steps towards artificial cognition,” in *IEEE International Conference on Robotics and Automation (ICRA’03)*, pp. 3140–3145, 2003.

[75] R. Detry, D. Kraft, A. Buch, N. Kruger, and J. Piater, “Refining grasp affordance models by experience,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2287 –2293, may 2010.

[76] E. Ugur, E. Şahin, and E. Oztop, “Affordance learning from range data for multi-step planning,” *Int. Conf. on Epigenetic Robotics*, 2009.

[77] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "A computational model of object affordances," in *Advances In Cognitive Systems*, IET, 2009.

[78] I. Atil, N. Dag, S. Kalkan, and E. Şahin, "Affordances and emergence of concepts," in *Epigenetic Robotics*, 2010.

[79] N. Dag, I. Atil, S. Kalkan, and E. Sahin, "Learning affordances for categorizing objects and their properties," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 3089–3092, IEEE, 2010.

[80] J. Sinapov and A. Stoytchev, "Learning and generalization of behavior-grounded tool affordances," in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pp. 19 –24, july 2007.

[81] J. Sinapov and A. Stoytchev, "Detecting the functional similarities between tools using a hierarchical representation of outcomes," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, pp. 91 –96, aug. 2008.

[82] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *Robotics, IEEE Transactions on*, vol. 24, pp. 15–26, feb. 2008.

[83] L. Montesano, M. Lopes, F. Melo, A. Bernardino, and J. Santos-Victor, "A computational model of object affordances," *Advances in Cognitive Systems*, vol. 54, p. 258, 2009.

[84] M. Richardson, K. Marsh, and R. Baron, "Judging and actualizing intrapersonal and interpersonal affordances.," *Journal of experimental psychology: Human Perception and Performance*, vol. 33, no. 4, p. 845, 2007.

[85] K. Marsh, M. Richardson, and R. Schmidt, "Social connection through joint action and interpersonal coordination," *Topics in Cognitive Science*, vol. 1, no. 2, pp. 320–339, 2009.

[86] R. E. Bellman, "Adaptive control processes: A guided tour," 1961.

[87] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pp. 58–65, 1998.

[88] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A multi resolution clustering approach for very large spatial databases," in *Proceedings of the 24th VLDB Conference*, 1998.

[89] V. K. Michael Steinbach, Levent Ertöz, "The challenges of clustering high-dimensional data," *New Vistas in Statistical Physics Applications in Econophysics Bioinformatics and Pattern Recognition*, 2003.

[90] A. M. D. Pelleg, "X means: extending k means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734, 2000.

[91] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop on Machine Learning*, pp. 249–256, 1992.

[92] I. Kononenko *Machine Learning ECML94*, vol. 784, 1994.

[93] H. Frigui, N. Boujemaa, and S. ann Lim, "Unsupervised clustering and feature discrimination with application to image database categorization," in *In NAFIPS*, 2001.

[94] H. Zeng and Y. ming Cheung, "Feature selection for clustering on high dimensional data," in *PRICAI*, pp. 913–922, Springer, 2008.

[95] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[96] C. J. L. C C Chang, "Libsvm: a library for svm," pp. 1–30, 2001.

[97] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, 1990.

[98] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map," pp. 15–19, IEEE Computer Society, 2000.

[99] T. Martinetz and K. Schulten, "A neural gas network learns topologies," *Artificial Neural Networks*, pp. 397–402, 1991.

[100] G. M. L. Natale and G. Sandini, "Learning haptic representations of objects," in *Proceedings of the International Conference on Intelligent Manipulation and Grasping*, 2004.

[101] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pp. 50–56, 2008.

[102] P. F. Giorgio Metta and L. Natale, "Yarp: Yet another robot platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–48, 2006.

[103] L. N. P Fitzpatrick, G Metta, "Towards long-lived robot genes," vol. 56, pp. 29–45, 2008.

[104] Q. M., G. B., C. K., F. J., F. T., L. J., B. E., W. R., and N. A., "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[105] S. C. Radu Bogdan Rusu, "3d is here: Point cloud library (pcl)," *Library*, vol. 26, no. 2, pp. 1–4, 2011.

[106] R. M. Wee K Leow, "Priming, perceptual reversal, and circular reaction in a neural network model of schema-based vision," in *Proceedings of the 16th Annual Conference of the Cognitive Science Society*, pp. 560–565, 1994.

[107] I. Lerner, S. Bentin, M. Scopus, J. Israel, and O. Shriki, "Automatic and controlled processes in semantic priming : an attractor neural network model with latching dynamics," pp. 1112–1117, 1992.

[108] M. E. Howard Bowman, Friederike Schlaghecken, "A neural network model of inhibitory processes in subliminal priming," vol. 13, no. 4, 2006.

[109] A. K. Qin and P. N. Suganthan, "Robust growing neural gas algorithm with application in cluster analysis," *Neural Networks*, vol. 17, no. 8-9, pp. 1135–1148, 2004.

[110] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software," vol. 11, no. 1, 2009.

[111] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[112] P. Grünwald, *The Minimum Description Length Principle*. The MIT Press, 2007.

APPENDIX A

USED LEARNING ALGORITHMS

In this appendix chapter, we depict and briefly comment on our usage of known algorithms in the literature.

A.1 Relief and ReliefF Algorithms

The Relief algorithm has been first proposed by Kira and Rendell [91]. It is a feauture ranking (i.e. discrimination) algorithm, whose use can be extended for feature selection. Briefly, it evaluates each input according to its distance to the closest input from the same class (*nearest hit*) and from a different class (*nearest miss*). This evaluation reveals to what extent does the given input contribute to the characteristics that defines its class. Its extension, ReliefF, searches for k nearest neighbors from both same and different classes to obtain nearest hits and nearest misses, respectively. ReliefF can also deal with missing data, by using estimation methods. The pseudocode of Relief algorithm is sketched in Algorithm 3.

These two algorithms are reported to be weak in noisy and multi-class data, which is verified in our experiments. In our work, we have chosen to use the ReliefF implementation of WEKA [110], an open source software tool for knowledge analysis with a repertoire of very useful machine learning algorithms.

A.2 Self Organizing Maps - SOM

First proposed by Kohonen [97], therefore sometimes called as Kohonen Maps, Self-Organizing Maps (SOM) is an unsupervised variant of neural network algorithms. Its usage mainly lies

Algorithm 3 The Pseudo-code for Relief

Require: Input set $\mathcal{I}$

Require: Output vector $\mathcal{W}$ where $||\mathcal{W}|| = A$

```
for a = 1 → A do
    W_a ← 0
end for
for i = 1 → m do
    R ← random(I)
    H ← nearestHit(R)
    M ← nearestMiss(R)
    for a = 1 → A do
        W_a ← W_a − diff(a, R, H) ÷ m + diff(a, R, M) ÷ m
    end for
end for
```

on discretization of a input space. We utilize this algorithm to address a multi-class learning scheme, i.e., grouping the effect information into effect clusters.

In order to reflect our design choices, we have implemented our own version of SOM in MATLAB [111]. This version of ours is sketched in Algorithm 4.

A.3 Robust Growing Neural Gas - RGNG

Robust Growing Neural Gas is a variant of Neural Gas, which are briefly mentioned in Section 2.6. The algorithm was first introduced in [109]. Just like SOM, it employs a competitive learning scheme, where each input is assigned to the node whose weight vector is closest. Meanwhile, it uses the Minimum Description Length criteria, which is an important concept in information theory [112], to form new nodes until the optimum number of nodes is found.

In this thesis, we utilized this algorithm to be able to convey meaningful interpretations of robot's interactions. We have used a MATLAB implementation of this algorithm (Courtesy of Ilkay Atıl, who has also demonstrated the working of this algorithm [7]).

Algorithm 4 The modified self-organizing maps

$L \leftarrow randomPermutation(1 : effectCount)$

for all $w_i \epsilon \mathcal{W}$ **do**

  $w_i^{language} \leftarrow L_i$

  for all $j \neq language$ **do**

    $w_i^j \leftarrow random(0..1)$

  end for

end for

for $e = 1 \rightarrow numberOfEpochs$ **do**

  for $n = 1 \rightarrow numberOfInputs$ **do**

    - $v \leftarrow nextItem$

    - Compute the winner node according to the distance metrics discussed in Section 3.4.1:

    $winner \leftarrow \arg\max_i dist(w_i, v)$

    - Get the neighbourhood of w_{winner}, i.e. $\mathcal{N} \subseteq \mathcal{W}$ $\mathcal{N} \leftarrow neighbors(w_{winner})$

    - Update the winner node and its neighbours according to update functions discussed in Section 3.4.1:

    for all $w_w \epsilon N$ **do**

      $w_w \leftarrow w_w + \alpha \times y_i \times dist(w_w, v)$

    end for

  end for

end for

APPENDIX B

SOFTWARE PLATFORMS

In this appendix chapter, we review the software platforms used in our work.

B.1 YARP - Yet Another Robotic Platform

YARP [102] is an open source software that provides the network backbone of robotic systems. In a typical YARP network, there exists a single server, to which all modules register by unique port names. Message passing between YARP modules are typically done through Bottle objects, which are essentially containers that can accomodate multiple types of data. YARP also supports online image viewing through a module called YarpView.

Nearly all known modules for iCub work with YARP to communicate information. Therefore, this platform is an indispensable feature for our system.

B.2 ROS

ROS [104] is an open source software that contains many useful algorithms that range from perception to motion control to be used in robotic applications.

As put by Quigley et.al., ROS is not an operating system in the traditional sense of process management and scheduling; rather, it provides a structured communications layer above the host operating systems of a heterogenous compute cluster.

In contrast to YARP modules that communicate via ports, ROS modules communicate on topics that are to declared and accounted within the central control module called roscore.

ROS is especially useful in implementing a service-oriented system, which is, in author's opinion, vital for any complex robotic system.

B.3 PCL - Point Cloud Library

PCL [105] is an open source library dedicated for processing information on point clouds. Most of the time, this point clouds contain distance and coordinate values received from the camera. The library contains many useful algorithms for feature extraction as well, such as principle curvature and surface normal estimations. Originally a part of ROS [104], it is now developed as a separate project.