Proceedings

AIC 2013

1st International Workshop on

Artificial Intelligence and Cognition

Edited by
Antonio Lieto
Marco Cruciani

December 3rd, 2013, Torino, Italy
A workshop of AI*IA 2013 - 25th Year Anniversary

Preface

This book of Proceedings contains the accepted papers of the first International Workshop on Artificial Intelligence and Cognition (AIC13). The workshop, held in Turin (Italy) on 3rd December 2013, has been co-located with the XIII International Conference of the Italian Association on Artificial Intelligence.

The scientific motivation behind AIC13 resides on the growing impact that, in the last years, the collaboration between Cognitive Science and Artificial Intelligence (AI) had for both the disciplines. In AI this partnership has driven to the realization of intelligent systems based on plausible models of human cognition. In turn, in cognitive science, the partnership allowed the development of cognitive models and architectures (based on information processing, on representations and their manipulation, etc.) providing greater understanding on human thinking.

The spirit and aim of the AI and Cognition workshop is therefore that one of putting together researchers coming from different domains (e.g., artificial intelligence, cognitive science, computer science, engineering, philosophy, social sciences, etc.) working on the interdisciplinary field of cognitively inspired artificial systems. In this workshop proceedings appear 2 abstracts of the talks provided by the keynote speakers and 16 peer reviewed papers. Specifically 8 full papers (31 % acceptance rate) and 8 short papers were selected on a total of 26 submissions coming from researchers of 14 different countries.

In the following a short introduction to the content of the papers (full and short) is presented.

In the paper "Simulating Actions with the Associative Self-Organizing Map" by Miriam Buonamente, Haris Dindo, Magnus Johnsson, the authors present a method based on the Associative Self Organizing Map (A-SOM) used for learning and recognizing actions. The authors show how their A-SOM based systems, once learnt to recognize actions, uses this learning to predict the continuation of an observed initial movement of an agent, predicting, in this way, its intentions.

In the paper "Acting on Conceptual Spaces in Cognitive Agents" by Agnese Augello, Salvatore Gaglio, Gianluigi Oliveri, Giovanni Pilato, the authors discuss the idea of providing a cognitive agent, whose conceptual representations are assumed to be grounded on the conceptual spaces framework (CS), with the ability of producing new spaces by means of global operations. With this goal in mind two operations on the Conceptual Spaces framework are proposed.

In the paper "Using Relational Adjectives for Extracting Hyponyms from Medical Texts" by Olga Acosta, Cesar Aguilar and Gerardo Sierra, the authors expose a method for extracting hyponyms and hyperonyms from analytical definitions, focusing on the relation observed between hyperonyms and relational adjectives. For detecting the hyperonyms associated to relational adjectives, they used a set of linguistic heuristics applied in medical texts in Spanish.

In the paper "Controlling a General Purpose Service Robot By Means Of a Cognitive Architecture" by Jordi-Ysard Puigbo, Albert Pumarola and Ricardo Tellez, the authors present a humanoid service robot equipped with a set of simple action skills including navigating, grasping, recognizing objects or people, etc. The robot has to complete a voice command in natural language that encodes a complex task. To decide which of those skills should be activated and in which sequence the SOAR cognitive architecture has been used. SOAR acts as a reasoner that selects the current action the robot must do, moving it towards the goal. The architecture allows to include new goals by just adding new skills.

In the paper "Towards a Cognitive Architecture for Music Perception" by Antonio Chella, the author presents a framework of a cognitive architecture for music perception. The architecture takes into account many relationships between vision and music perception and its focus resides in the intermediate area between the subsymbolic and the linguistic areas, based on conceptual spaces. Also, a conceptual space for the perception of notes and chords is discussed, and a focus of attention mechanism scanning the conceptual space is outlined.

In the paper "Typicality-Based Inference by Plugging Conceptual Spaces Into Ontologies" by Leo Ghignone, Antonio Lieto and Daniele P. Radicioni the authors propose a cognitively inspired system for the representation of conceptual information in an ontology-based environment. The authors present a system designed to provide a twofold view on the same artificial concept combining a classic symbolic component (grounded on a formal ontology) with a typicality-based one (grounded on the Conceptual Spaces framework). The implemented system has been tested in a pilot experimentation regarding the classification task of linguistic stimuli.

In the paper "Introducing Sensory-motor Apparatus in Neuropsychological Modelization" by Onofrio Gigliotta, Paolo Bartolomeo and Orazio Miglino, the authors present artificial embodied neural agents equipped with a pan/tilt camera, provided with different neural and motor capabilities, to solve a well known neuropsychological test: the cancellation task. The paper shows that embodied agents provided with additional motor capabilities (a zooming motor) outperform simple pan/tilt agents even when controlled by more complex neural controllers.

In the paper "How Affordances can Rule the (Computational) World" by Alice Ruggeri and Luigi Di Caro, the authors propose the idea of integrating the concept of affordance within the ontology based representations. The authors propose to extend the idea of ontologies taking into account the subjectivity of the agents that are involved in the interaction with an external environment. Instead of duplicating objects, according to the interaction, the ontological representations should change their aspects, fitting the specific situations that take place. The authors suggest that this approach can be used in different domains from Natural Language Processing techniques and Ontology Alignment to User Modeling.

In the paper "Latent Semantic Analysis as Method for Automatic Question Scoring" by David Tobinski and Oliver Kraft, the authors discuss the rating

of one item taken from an exam using Latent Semantic Analysis (LSA). It is attempted to use documents in a corpus as assessment criteria and to project student answers as pseudo-documents into the semantic space. The paper shows that as long as each document is sufficiently distinct from each other, it is possible to use LSA to rate open questions.

In the paper "Higher-order Logic Description of MDPs to Support Meta-cognition in Artificial Agents" by Roberto Pirrone, Vincenzo Cannella and Antonio Chella, the authors propose a formalism to represent factored MDPs in higher- order logic. This work proposes a mixed representation that combines both numerical and propositional formalism to describe Algebraic Decision Diagrams (ADDs) using first-, second- and third-order logic. In this way, the MDP description and the planning processes can be managed in a more abstract manner. The presented formalism allows manipulating structures, which describe entire MDP classes rather than a specific process.

In the paper Dual Aspects of Abduction and Induction by Flavio Zelazek, the author proposes a new characterization of abduction and induction based on the idea that the various aspects of the two kinds of inference rest on the essential features of increment of comprehension and extension of the terms involved. These two essential features are in a reciprocal relation of duality, whence the highlighting of the dual aspects of abduction and deduction.

In the paper "Plasticity and Robotics" by Martin Flament Fultot, the author focuses on the link between robotic systems and living systems, and sustains that behavioural plasticity constitutes a crucial property that robots must share with living beings. The paper presents a classification of the different aspects of plasticity that can contribute to a global behavioral plasticity in robotic and living systems.

In the paper "Characterising Ctations in Scholarly Articles: an Experiment" by Paolo Ciancarini, Angelo Di Iorio, Andrea Giovanni Nuzzolese, Silvio Peroni and Fabio Vitali, the authors present some experiments in letting humans annotate citations according to the CiTO ontology, a OWL-based ontology for describing the nature of citations, and compare the performance of different users.

In the paper "A Meta-Theory for Knowledge Representation" by Janos Sarbo, the author faces the problem of representation of meaningful interpretations in AI. He sustains that a process model of cognitive activities can be derived from the Peircean theory of categories, and that this model may function as a meta-theory for knowledge representation, by virtue of the fundamental nature of categories.

In the paper "Linguistic Affordances: Making Sense of Word Senses" by Alice Ruggeri and Luigi Di Caro, the authors focus the attention on the roles of word senses in standard Natural Language Understanding tasks. They propose the concept of linguistic affordances (i.e., combinations of objects properties that are involved in specific actions and that help the comprehension of the whole scene being described), and argue that similar verbs involving similar properties of the arguments may refer to comparable mental scenes.

In the paper "Towards a Formalization of Mental Model Reasoning for Syllogistic Fragments" by Yutaro Sugimoto, Yuri Sato and Shigeyuki Nakayama, the authors consider the recent developments in implementations of mental models theory, and formulate a mental model of reasoning for syllogistic fragments satisfying the formal requirements of mental model definition.

Acknowledgements

We would like to thank the keynote speakers of the workshop: Prof. Christian Freksa (University of Bremen, Germany) and Prof. Orazio Miglino (University of Napoli Federico II and ISTC-CNR, Italy) for accepting our invitation.

We sincerely thank the Interaction Models Group of the University of Turin, Italy (http://www.di.unito.it/ gull/), the Italian Association for Artificial Intelligence (AI*IA, http://www.aixia.it/), and the Italian Association of Cognitive Sciences (AISC, http://www.aisc-net.org) for their support in the organization of the workshop, and also the Rosselli Foundation (Fondazione Rosselli), Turin, Italy, for its logistic support.

We would like also to thank the members of the Scientific Committee for their valuable work during the reviewing process and the additional reviewers.

We would like to dedicate this book of proceedings to the Prof. Leonardo Lesmo, unfortunately no longer among us, that strongly encouraged and helped us in all the phases of the organization of this workshop.

December 2013
Antonio Lieto and Marco Cruciani
AIC 2013 Chairs

Organization

Organizers

Antonio Lieto	University of Torino, Italy
Marco Cruciani	University of Trento, Italy

Program Committee

Bruno Bara	University of Torino, Italy
Cristiano Castelfranchi	ISTC-CNR, Italy
Rosaria Conte	ISTC-CNR, Italy
Roberto Cordeschi	University La Sapienza, Italy
David Danks	Carnegie Mellon University, USA
Roberta Ferrario	LOA-ISTC, CNR, Italy
Marcello Frixione	University of Genova, Italyl
Francesco Gagliardi	University La Sapienza, Italy
Salvatore Gaglio	University of Palermo and ICAR-CNR, Italy
Aldo Gangemi	ISTC-CNR, Italy, and LIPN University Paris13-CNRS, France
Onofrio Gigliotta	University of Napoli Federico II , Italy
Ismo Koponen	University of Helsinki, Finland
Othalia Larue	University of Quebec a Montreal, Canada
Leonardo Lesmo	University of Torino, Italy
Ignazio Licata	School of Advanced Studies on Theoretical Methodologies of Physics, Italy
Diego Marconi	University of Torino, Italy
Orazio Miglino	University of Napoli Federico II, Italy
Alessandro Oltramari	Carnegie Mellon University, USA
Fabio Paglieri	ISTC-CNR, Italy
Pietro Perconti	University of Messina, Italy
Alessio Plebe	University of Messina, Italy
Daniele P. Radicioni	University of Turin, Italy
Alessandro Saffiotti	Orebro University, Sweden
Marco Tabacchi	University of Palermo, Italy
Pietro Terna	University of Torino, Italy

Additional Reviewers

Cristina Battaglino
Manuela Sanguinetti

Table of Contents

The Power of Space and Time: How Spatial and Temporal Structures Can Replace Computational Effort

Christian Freksa

University of Bremen, Germany, Cognitive Systems Group,
freksa@informatik.uni-bremen.de

Abstract. Spatial structures determine the ways we perceive our environment and the ways we act in it in important ways. Spatial structures also determine the ways we think about our environment and how we solve spatial problems abstractly. When we use graphics to visualize certain aspects of spatial and non-spatial entities, we exploit the power of spatial structures to better understand important relationships. We also are able to imagine spatial structures and to apply mental operations to them. Similarly, the structure of time determines the course of events in cognitive processing. In my talk I will present knowledge representation research in spatial cognition. I will demonstrate the power of spatial structures in comparison to formal descriptions that are conventionally used for spatial problem solving in computer science. I suggest that spatial and temporal structures can be exploited for the design of powerful spatial computers. I will show that spatial computers can be particularly suitable and efficient for spatio-temporal problem solving but may also be used for abstract problem solving in non-spatial domains.

When Psychology and Technology Converge. The Case of Spatial Cognition

Orazio Miglino

Natural and Artificial Cognition Lab, Department of Humanities, University of Naples Federico II, www.nac.unina.it, orazio.miglino@unina.it

Abstract. The behaviors of spatial orientation that an organism displays result from its capacity for adapting, knowing, and modifying its environment; expressed in one word, spatial orientation behaviors result from its psychology. These behaviors can be extremely simple (consider, for example, obstacle avoidance, tropisms, taxis, or random walks) but extremely sophisticated as well: consider for example, intercontinental migrations, orienting in tangled labyrinths, reaching unapproachable areas. In different species orienting abilities can be innate or the result of a long learning period in which teachers can be involved. This is the case for many vertebrates. Moreover, an organism can exploit external resources that amplify its exploring capacities; it can rely on others help and in this case what we observe is a sophisticated collective orienting behavior. An organism can use technological devices as well. Human beings have widely developed these two strategies - namely either exploring its own capacities or learning new orienting skills - and thanks to well-structured work groups (a crew navigating a boat, for instance) and the continuous improving of technological devices (geographical maps, satellites, compasses, etc.), they have expanded their habitat and can easily orient in skies and seas. It also is possible to observe orienting behaviors in an apparently paradoxical condition: exploring a world without moving ones body. In the present day a lot of interactions between humans and information and communication technologies (mobile phones, PCs, networks) are achieved using orienting behaviors. The best example is the World Wide Web: the explorer in this pure-knowledge universe navigates while keeping his/her body almost completely still. Spatial orientation behaviors are the final and observable outcome of a long chain made up by very complex psychobiological states and processes. There is no orienting without perception, learning, memory, motivation, planning, decision making, problem solving, and, in some cases, socialization. Explaining how an organism orients in space requires study of all human and animal cognition dimensions and, for this reason, psychology, and in more recent years anthropology, ethology, neuroscience all consider orientation a very interesting field of study. Bulding-up artificial systems (digital agents, simulated and physical robots, etc.) that shows the (almost) same behaviors of natural organisms is a powerful approach to reach a general theory of (spatial) cognition. In this framework the artificial systems could be viewed as new synthetic organisms to be behavioural compared with biological systems. On the other hand,

this approach could produce more adaptive and efficient systems artificial systems (such as autonomous mobile robots). I will present different experiments in Evolutionary Robotics designed to explain spatial cognition at different level of complexity (from avoiding behaviours to detour behaviours). Finally, I will try to delineate some general principles to building-up adaptive mobile agents.

Simulating Actions with the Associative Self-Organizing Map

Miriam Buonamente[1], Haris Dindo[1], and Magnus Johnsson[2]

[1] RoboticsLab, DICGIM, University of Palermo,
Viale delle Scienze, Ed. 6, 90128 Palermo, Italy
{miriam.buonamente,haris.dindo}@unipa.it
http://www.unipa.it
[2] Lund University Cognitive Science,
Lundagård, 222 22 Lund, Sweden
magnus@magnusjohnsson.se
http://www.magnusjohnsson.se

Abstract. We present a system that can learn to represent actions as well as to internally simulate the likely continuation of their initial parts. The method we propose is based on the Associative Self Organizing Map (A-SOM), a variant of the Self Organizing Map. By emulating the way the human brain is thought to perform pattern recognition tasks, the A-SOM learns to associate its activity with different inputs over time, where inputs are observations of other's actions. Once the A-SOM has learnt to recognize actions, it uses this learning to predict the continuation of an observed initial movement of an agent, in this way reading its intentions. We evaluate the system's ability to simulate actions in an experiment with good results, and we provide a discussion about its generalization ability. The presented research is part of a bigger project aiming at endowing an agent with the ability to internally represent action patterns and to use these to recognize and simulate others behaviour.

Keywords: Associative Self-Organizing Map, Neural Network, Action Recognition, Internal Simulation, Intention Understanding

1 Introduction

Robots are on the verge of becoming a part of the human society. The aim is to augment human capabilities with automated and cooperative robotic devices to have a more convenient and safe life. Robotic agents could be applied in several fields such as the general assistance with everyday tasks for elderly and handicapped enabling them to live independent and comfortable lives like people without disabilities. To deal with such desire and demand, natural and intuitive interfaces, which allow inexperienced users to employ their robots easily and safely, have to be implemented.

Efficient cooperation between humans and robots requires continuous and complex intention recognition; agents have to understand and predict human

intentions and motion. In our daily interactions, we depend on the ability to understand the intent of others, which allows us to read other's mind. In a simple dance, two persons coordinate their steps and their movements by predicting subliminally the intentions of each other. In the same way in multi-agents environments, two or more agents that cooperate (or compete) to perform a certain task have to mutually understand their intentions.

Intention recognition can be defined as the problem of inferring an agent's intention through the observation of its actions. This problem has been faced in several fields of human-robot collaboration [1]. In robotics, intention recognition has been addressed in many contexts like social interaction [2] and learning by imitation [3] [4] [5].

Intention recognition requires a wide range of evaluative processes including, among others, the decoding of biological motion and the ability to recognize tasks. This decoding is presumably based on the internal simulation [6] of other peoples behaviour within our own nervous system. The visual perception of motion is a particularly crucial source of sensory input. It is essential to be able to pick out the motion to predict the actions of other individuals. Johansson's experiment [7] showed that humans, just by observing points of lights, were able to perceive and understand movements. By looking at biological motion, such as Johansson's walkers, humans attribute mental states such as intentions and desires to the observed movements. Recent neurobiological studies [8] corroborate Johansson's experiment by arguing that the human brain can perceive actions by observing only the human body poses, called postures, during action execution. Thus, actions can be described as sequences of consecutive human body poses, in terms of human body silhouettes [9] [10] [11]. Many neuroscientists believe that the ability to understand the intentions of other people just by observing them depends on the so-called mirror-neuron system in the brain [12], which comes into play not only when an action is performed, but also when a similar action is observed. It is believed that this mechanism is based on the internal simulation of the observed action and the estimation of the actor's intentions on the basis of a representation of ones own intentions [13].

Our long term goal is to endow an agent with the ability to internally represent motion patterns and to use these patterns to recognize and simulate other's behaviour. The study presented here is part of a bigger project whose first step was to efficiently represent and recognize human actions [14] by using the Associative Self-Organizing Map (A-SOM) [15]. In this paper we want to use the same biologically-inspired model to predict an agent's intentions by internally simulating the behaviour likely to follow initial movements. As humans do effortlessly, agents have to be able to elicit the likely continuation of the observed action even if an obstacle or other factors obscure their view. Indeed, as we will see below, the A-SOM can remember perceptual sequences by associating the current network activity with its own earlier activity. Due to this ability, the A-SOM could receive an incomplete input pattern and continue to elicit the likely continuation, i.e. to carry out sequence completion of perceptual activity over time.

We have tested the A-SOM on simulation of observed actions on a suitable dataset made of images depicting the only part of the persons body involved in the movement. The images used to create this dataset was taken from the "INRIA 4D repository [3]", a publicly available dataset of movies representing 13 common actions: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, and throw (see Fig. 1).

This paper is organized as follows: A short presentation of the A-SOM network is given in section II. Section III presents the method and the experiments for evaluating performance. Conclusions and future works are outlined in section IV.

2 Associative Self-Organizing Map

The A-SOM is an extension of the Self-Organizing Map (SOM) [16] which learns to associate its activity with the activity of other neural networks. It can be considered a SOM with additional (possibly delayed) ancillary input from other networks, Fig. 2.

Ancillary connections can also be used to connect the A-SOM to itself, thus associating its activity with its own earlier activity. This makes the A-SOM able to remember and to complete perceptual sequences over time. Many simulations prove that the A-SOM, once receiving some initial input, can continue to elicit the likely following activity in the nearest future even though no further input is received [17] [18].

The A-SOM consists of an $I \times J$ grid of neurons with a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with $r+1$ weight vectors $w_{ij}^a \in R^n$ and $w_{ij}^1 \in R^{m_1}$, $w_{ij}^2 \in R^{m_2}$, ..., $w_{ij}^r \in R^{m_r}$. All the elements of all the weight vectors are initialized by real numbers randomly selected from a uniform distribution between 0 and 1, after which all the weight vectors are normalized, i.e. turned into unit vectors.

At time t each neuron n_{ij} receives $r+1$ input vectors $x^a(t) \in R^n$ and $x^1(t-d_1) \in R^{m_1}$, $x^2(t-d_2) \in R^{m_2}$, ..., $x^r(t-d_r) \in R^{m_r}$ where d_p is the time delay for input vector x^p, $p = 1, 2, \ldots, r$.

The main net input s_{ij} is calculated using the standard cosine metric

$$s_{ij}(t) = \frac{x^a(t) \cdot w_{ij}^a(t)}{||x^a(t)||||w_{ij}^a(t)||}, \tag{1}$$

The activity in the neuron n_{ij} is given by

$$y_{ij} = [y_{ij}^a(t) + y_{ij}^1(t) + y_{ij}^2(t) + \ldots + y_{ij}^r(t)]/(r+1) \tag{2}$$

where the main activity y_{ij}^a is calculated by using the softmax function [19]

[3] The repository is available at http://4drepository.inrialpes.fr. It offers several movies representing sequences of actions. Each video is captured from 5 different cameras. For the experiments in this paper we chose the movie "Julien1" with the frontal camera view "cam0".

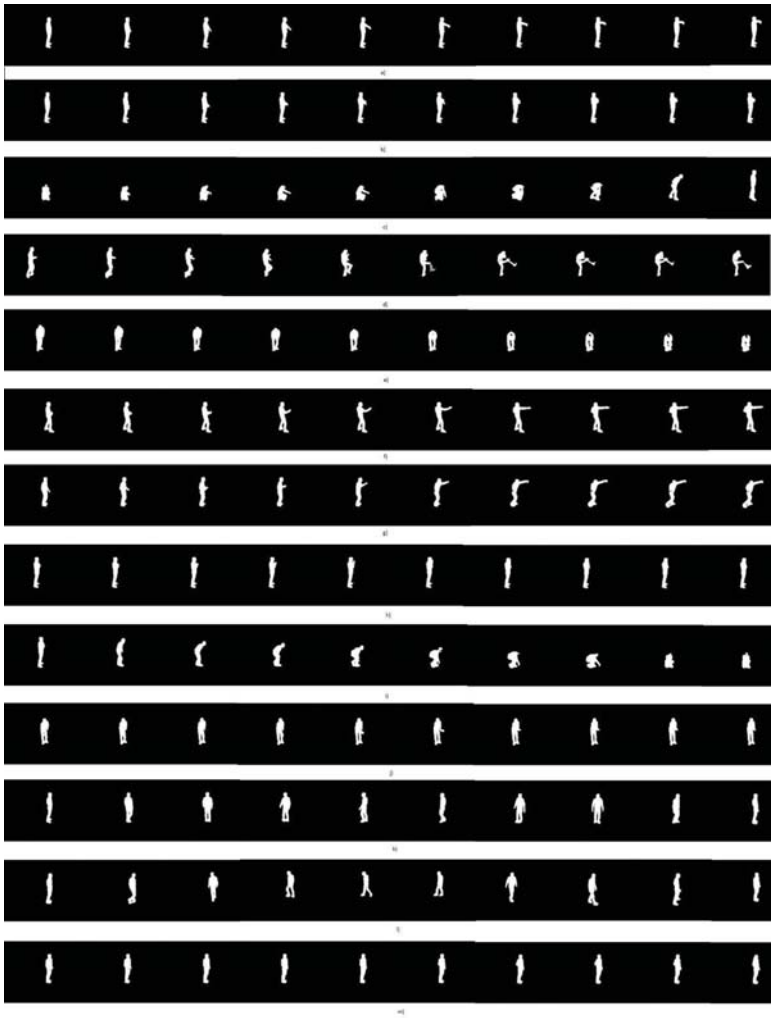

Fig. 1. Prototypical postures of 13 different actions in our dataset: check watch, cross arms, get up, kick, pick up, point, punch, scratch head, sit down, throw, turn around, walk, wave hand.

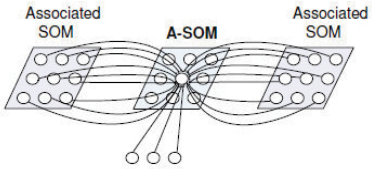

Fig. 2. An A-SOM network connected with two other SOM networks. They provide the ancillary input to the main A-SOM (see the main text for more details).

$$y_{ij}^{a}(t) = \frac{(s_{ij}(t))^m}{\max_{ij}(s_{ij}(t))^m} \tag{3}$$

where m is the softmax exponent.

The ancillary activity $y_{ij}^{p}(t)$, p=1,2,...,r is calculated by again using the standard cosine metric

$$y_{ij}^{p}(t) = \frac{x^p(t-d_p)\cdot w_{ij}^{p}(t)}{||x^p(t-d_p)||||w_{ij}^{p}(t)||}. \tag{4}$$

The neuron c with the strongest main activation is selected:

$$c = \text{argmax}_{ij} y_{ij}(t) \tag{5}$$

The weights w_{ijk}^{a} are adapted by

$$w_{ijk}^{a}(t+1) = w_{ijk}^{a}(t) + \alpha(t)G_{ijc}(t)[x_k^a(t) - w_{ijk}^{a}(t)] \tag{6}$$

where $0 \leq \alpha(t) \leq 1$ is the adaptation strength with $\alpha(t) \to 0$ when $t \to \infty$. The neighbourhood function $G_{ijc}(t) = e^{-\frac{||r_c - r_{ij}||}{2\sigma^2(t)}}$ is a Gaussian function decreasing with time, and $r_c \in R^2$ and $r_{ij} \in R^2$ are location vectors of neurons c and n_{ij} respectively.

The weights w_{ijl}^{p}, p=1,2,...,r, are adapted by

$$w_{ijl}^{p}(t+1) = w_{ijl}^{p}(t) + \beta x_l^p(t-d_p)[y_{ij}^{a}(t) - y_{ij}^{p}(t)] \tag{7}$$

where β is the adaptation strength.

All weights $w_{ijk}^{a}(t)$ and $w_{ijl}^{p}(t)$ are normalized after each adaptation.

In this paper the ancillary input vector x^1 is the activity of the A-SOM from the previous iteration rearranged into a vector with the time delay $d_1 = 1$.

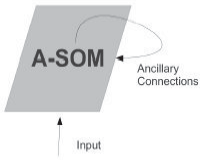

Fig. 3. The model consisting of an A-SOM with time-delayed ancillary connections connected to itself.

3 Experiment

We want to evaluate if the bio-inspired model, introduced and tested for the action recognition task in [14], Fig. 3, is also able to simulate the continuation of the initial part of an action. To this end, we tested the simulation capabilities of the A-SOM. The experiments scope is to verify if the network is able to receive an incomplete input pattern and continue to elicit the likely continuation of recognized actions. Actions, defined as single motion patterns performed by a single human [20], are described as sequences of body postures.

The dataset of actions is the same as we used for the recognition experiment in [14]. It consists of more than 700 postural images representing 13 different actions. Since we want the agent to be able to simulate one action at a time, we split the original movie into 13 different movies: one movie for each action (see Fig. 1). Each frame is preprocessed to reduce the noise and to improve its quality and the posture vectors are extracted (see section 3.1 below). The posture vectors are used to create the training set required to train the A-SOM. Our final training set is composed of about 20000 samples where every sample is a posture vector.

The created input is used to train the A-SOM network. The training lasted for about 90000 iterations. The generated weight file is used to execute tests. The implementation of all code for the experiments presented in this paper was done in C++ using the neural modelling framework Ikaros [21]. The following sections detail the preprocessing phase as well as the results obtained.

3.1 Preprocessing phase

To reduce the computational load and to improve the performance, movies should have the same duration and images should depict the only part of the body involved in the movement. By reducing the numbers of images for each movie to 10, we have a good compromise to have seamless and fluid actions, guaranteeing the quality of the movie. As Fig. 4 shows, the reduction of the number of images, depicting the "walk action" movie, does not affect the quality of the action reproduction.

Consecutive images were subtracted to depict the only part of the body involved in the action, focusing in this way the attention on the movement ex-

Fig. 4. The walk action movie created with a reduced number of images.

Fig. 5. a) The sequence of images depicting the check watch action; b) The sequence of images obtained by subtracting consecutive images of the check watch action.

clusively. This operation further reduced the number of frames for each movie to 9, without affecting the quality of the video. As can be seen in Fig. 5, in the "walk action" only the arm is involved in the movement.

To further improve the system's performance, we need to produce binary images of fixed and small size. By using a fixed boundary box, including the part of the body performing the action, we cut out the images eliminating anything not involved in the movement. In this way, we simulate an attentive process in which the human eye observes and follows the salient parts of the action only. To have smaller representations the binary images depicting the actions were shrunk to 30×30 matrices. Finally, the obtained matrix representations were vectorized to produce 9 posture vectors $p \in R^{\mathrm{D}}$, where $D = 900$, for each action. These posture vectors are used as input to the A-SOM.

3.2 Action Simulation

The objective was to verify whether the A-SOM is able to internally simulate the likely continuation of initial actions. Thus, we fed the trained A-SOM with incomplete input patterns and expected it to continue to elicit activity patterns corresponding to the remaining part of the action. The action recognition task has been already tested in [14] with good results. The system we set up was the same as the one used in [14] and consists of one A-SOM connected to itself with time delayed ancillary connections. To evaluate the A-SOM, 13 sequences each containing 9 posture vectors were constructed as explained above. Each of these sequences represents an action. The posture vectors represent the binary images that form the videos and depict only the part of the human body involved in the action, see Fig.6

We fed the A-SOM with one sequence at a time, reducing the number of posture vectors at the end of the sequence each time and replacing them with

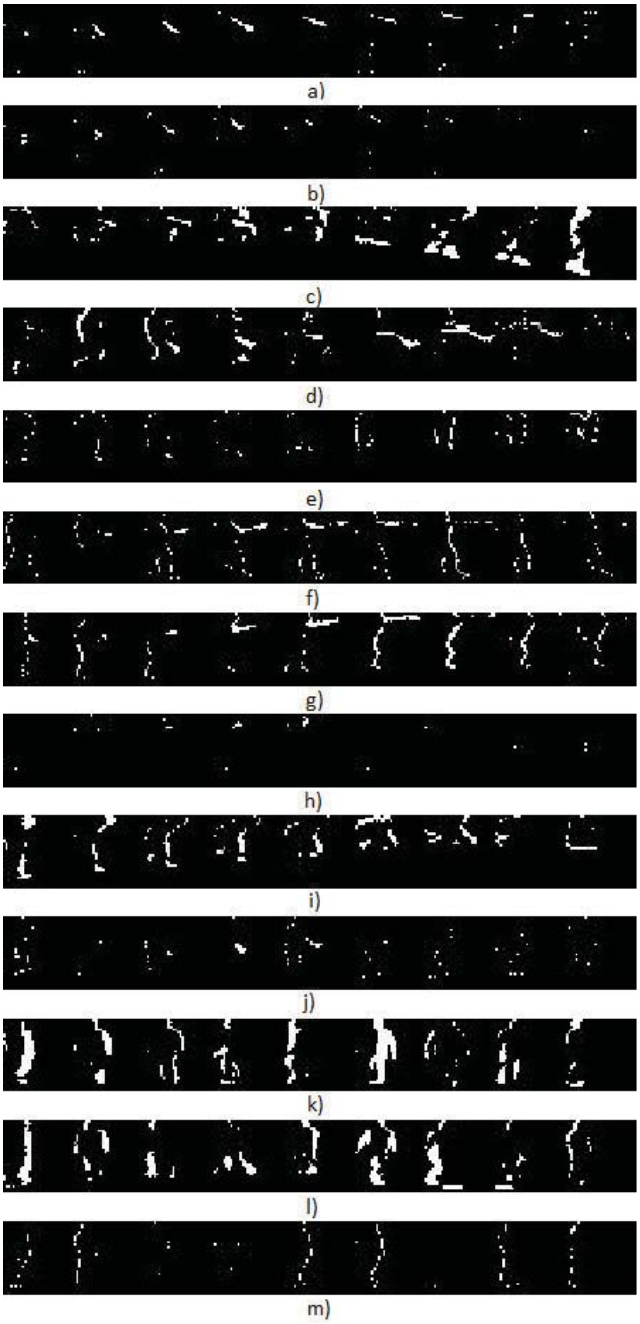

Fig. 6. The parts of the human body involved in the movement of each action. Each sequence was obtained by subtracting consecutive images in each movie. The actions are: a) check watch; b) cross arm; c) get up; d) kick; e) pick up; f) point; g) punch; h) scratch head; i) sit down; j) throw; k) turn around; l) walk; m) wave hand.

null vectors (representing no input). In this way, we created the incomplete input that the A-SOM has to complete.The conducted experiment consisted of several tests. The first one was made by using the sequences consisting of all the 9 frames with the aim to record the coordinates of the activity centres generated by the A-SOM and to use these values as reference values for the further iterations. Subsequent tests had the sequences with one frame less (replaced by a null vector representing no input) each time and the A-SOM had the task to complete the frame sequence by eliciting activity corresponding to the activity representing the remaining part of the sequence. The last test included only the sequences made of one frame (followed by 8 null vectors representing no input).

The centres of activity generated by the A-SOM at each iteration were collected in tables, and colour coding was used to indicate the ability (or the inability) of the A-SOM to predict the action continuation. The dark green colour indicates that the A-SOM predicted the right centres of activity; the light green indicates that the A-SOM predicted a value close to the expected centre of activity and the red one indicates that the A-SOM could not predict the right value, see Fig.7. The ability to predict varies with the type of action. For actions like "sit down" and "punch", A-SOM needed 8 images to predict the rest of the sequence; whereas for the "walk" action, A-SOM needed only 4 images to complete the sequence. In general the system needed between 4 and 9 inputs to internally simulate the rest of the actions. This is a reasonable result, since even humans cannot be expected to be able to predict the intended action of another agent without a reasonable amount of initial information. For example, looking at the initial part of an action like "punch", we can hardly say what the person is going to do. It could be "punch" or "point"; we need more frames to exactly determine the performed action. In the same way, looking at a person starting to walk, we cannot say in advance if the person would walk or turn around or even kick because the initial postures are all similar to one another.

The results obtained through this experiment allowed us to speculate about the ability of the A-SOM to generalize. The generalization is the network's ability to recognize inputs it has never seen before. Our idea is that if the A-SOM is able to recognize images as similar by generating close or equal centres of activity, then it will also be able to recognize an image it has never encountered before if this is similar to a known image. We checked if similar images had the same centres of activity and if similar centres of activity corresponded to similar images. The results show that the A-SOM generated very close or equal values for very similar images, see Fig.8. Actions like "turn around", "walk" and "get up" present some frames very similar to each other and for such frames the A-SOM generates the same centres of activity. This ability is validated through the selection of some centres of activity and the verification that they correspond to similar images. "Check watch", "get up", "point" and "kick" actions include in their sequences frames depicting the movement of the arm that can be attributed to all of them. For these frames the A-SOM elicits the same centre of activity, see Fig. 9. The results presented here support the belief that our system is also able to generalize.

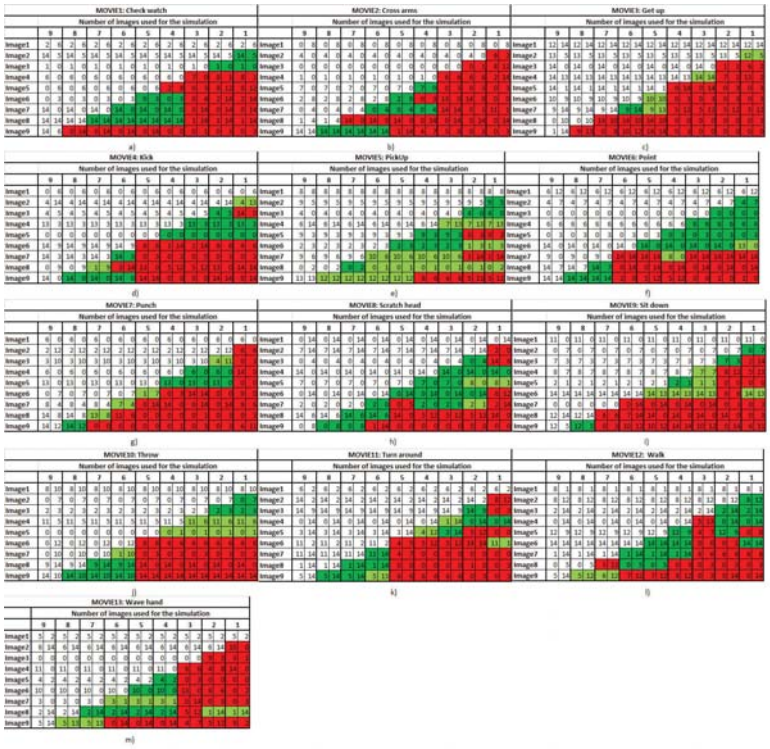

Fig. 7. Simulation results: The tables show the ability of the A-SOM to continue the likely continuation of an observed behaviour. Dark green colour indicates that the A-SOM is able to simulate, light green colour indicates that the A-SOM predicts a value very close to the expected one, and red colour indicates that the A-SOM predicts the wrong value. The system needs between 4 and 9 inputs to internally simulate the rest of the sequence.

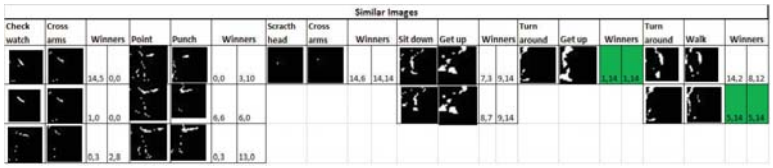

Fig. 8. Similar images have similar centres of activity. The A-SOM elicits similar or equal centres of activity for images that are similar.

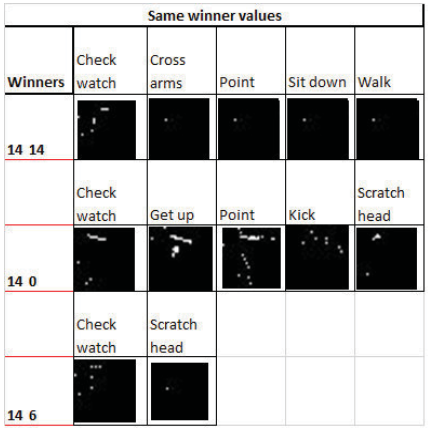

Fig. 9. Images with the same centres of activity (winners). The frames present similar features which lead the A-SOM to elicit the same centre of activity.

4 Conclusion

In this paper, we proposed a new method for internally simulating behaviours of observed agents. The experiment presented here is part of a bigger project whose scope is to develop a cognitive system endowed with the ability to read other's intentions. The method is based on the A-SOM, a novel variant of the SOM, whose ability of recognition and classification has already been tested in [14]. In our experiment, we connected the A-SOM to itself with time delayed ancillary connections and the system was trained and tested with a set of images depicting the part of the body performing the movement. The results presented here show that the A-SOM can receive some initial sensory input and internally simulate the rest of the action without any further input.

Moreover, we verified the ability of the A-SOM to recognize input never encountered before, with encouraging results. In fact, the A-SOM recognizes similar actions by eliciting close or identical centres of activity.

We are currently working on improving the system to increase the recognition and simulation abilities.

Acknowledgements The authors gratefully acknowledge the support from the Linnaeus Centre Thinking in Time: Cognition, Communication, and Learning, financed by the Swedish Research Council, grant no. 349-2007-8695.

References

1. Awais, M., Henrich, D.: Human-robot collaboration by intention recognition using probabilistic state machines. In: Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop on Robotics. (2010) 75–80
2. Breazeal, C.: Designing sociable robots. the MIT Press (2004)
3. Chella, A., Dindo, H., Infantino, I.: A cognitive framework for imitation learning. Robotics and Autonomous Systems **54**(5) (2006) 403–408
4. Chella, A., Dindo, H., Infantino, I.: Imitation learning and anchoring through conceptual spaces. Applied Artificial Intelligence **21**(4-5) (2007) 343–359
5. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems **57**(5) (2009) 396–483
6. Hesslow, G.: Conscious thought as simulation of behaviour and perception. Trends in Cognitive Sciences **6** (2002) 242–247
7. Johansson, G.: Visual perception of biological motion and a model for its analysis. Perception & Psychophysics **14**(2) (1973) 201–211
8. Giese, M.A., Poggio, T. Nat Rev Neurosci **4**(3) (March 2003) 179–192
9. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12) (2007) 2247–2253
10. Iosifidis, A., Tefas, A., Pitas, I.: View-invariant action recognition based on artificial neural networks. IEEE Trans. Neural Netw. Learning Syst. **23**(3) (2012) 412–424
11. Gkalelis, N., Tefas, A., Pitas, I.: Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. IEEE Transactions on Circuits Systems Video Technology **18**(11) (2008) 15111521
12. Rizzolatti, G., Craighero, L.: The mirror-neuron system. Annual Review of Neuroscience **27** (2004) 169192
13. Goldman, A.I.: Simulating minds: The philosophy, psychology, and neuroscience of mindreading. (2) (2006)
14. Buonamente, M., Dindo, H., Johnsson, M.: Recognizing actions with the associative self-organizing map. In: the proceedings of the XXIV International Conference on Information, Communication and Automation Technologies (ICAT 2013). (2013)
15. Johnsson, M., Balkenius, C., Hesslow, G.: Associative self-organizing map. In: Proceedings of IJCCI. (2009) 363–370
16. Kohonen, T.: Self-Organization and Associative Memory. Springer Verlag (1988)
17. Johnsson, M., Gil, D., Balkenius, C., Hesslow, G.: Supervised architectures for internal simulation of perceptions and actions. In: Proceedings of BICS. (2010)
18. Johnsson, M., Mendez, D.G., Hesslow, G., Balkenius, C.: Internal simulation in a bimodal system. In: Proceedings of SCAI. (2011) 173–182
19. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
20. Turaga, P.K., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. IEEE Trans. Circuits Syst. Video Techn. **18**(11) (2008) 1473–1488
21. Balkenius, C., Morén, J., Johansson, B., Johnsson, M.: Ikaros: Building cognitive models for robots. Advanced Engineering Informatics **24**(1) (2010) 40–48

Acting on Conceptual Spaces in Cognitive Agents

Agnese Augello[3], Salvatore Gaglio[1,3], Gianluigi Oliveri[2,3], and Giovanni Pilato[3]

[1] DICGIM- Università di Palermo
Viale delle Scienze, Edificio 6 - 90128, Palermo - ITALY
[2] Dipartimento di Scienze Umanistiche - Università di Palermo
Viale delle Scienze, Edificio 12 - 90128, Palermo - ITALY
[3] ICAR - Italian National Research Council
Viale delle Scienze - Edificio 11 - 90128 Palermo, Italy
{gaglio.oliveri}@unipa.it
{augello,pilato}@icar.pa.cnr.it

Abstract. Conceptual spaces were originally introduced by Gärdenfors as a bridge between symbolic and connectionist models of information representation. In our opinion, a cognitive agent, besides being able to work within his (current) conceptual space, must also be able to 'produce a new space' by means of 'global' operations. These are operations which, acting on a conceptual space taken as a whole, generate other conceptual spaces.

1 Introduction

The introduction of a cognitive architecture for an artificial agent implies the definition of a conceptual representation model. Conceptual spaces, used extensively in the last few years [1] [2] [3], were originally introduced by Gärdenfors as a bridge between symbolic and connectionist models of information representation. This was part of an attempt to describe what he calls the 'geometry of thought'.

If, for the sake of argument, we accept Gärdenfors paradigm of conceptual spaces, and intend to avoid the implausible idea that a cognitive agent comes with a potentially infinite library of conceptual spaces, we must conclude that a cognitive agent, besides being able to work within his (current) conceptual space, must also be able to 'produce a new space' by means of 'global' operations. These are operations which, acting on a conceptual space taken as a whole, generate other conceptual spaces.

We suppose that an agent acts like an experimenter: depending on the particular problem he has to solve, he chooses, either consciously or unconsciously, what to observe and what to measure. Both the environment and the internal state of the agent, which includes his intentions and goals, affect the manner in which the agent perceives, by directing the focus of its measurements on specific objects.

In this work we focus on operations that can be performed *in* and *on* conceptual spaces in order to allow a cognitive agent (CA) to produce his conceptual representation of the world according to his goals and his perceptions.

In the following sections, after a background on Conceptual Spaces theory, we introduce such operations and we discuss an example of the way they come to be applied in practice.

2 Conceptual spaces

In [4] and [5] we find a description of a cognitive architecture for modelling representations. This is a cognitive architecture in which an intermediate level, called 'geometric conceptual space', is introduced between a linguistic-symbolic level and an associationist sub-symbolic level of information representation.

According to the linguistic/symbolic level:

> Cognition is seen as essentially being *computation*, involving symbol manipulation. [4]

whereas, for the associationist sub-symbolic level:

> Associations among different kinds of information elements carry the main burden of representation. *Connectionism* is a special case of associationism that models associations using artificial neuron networks [4], where the behaviour of the network as a whole is determined by the initial state of activation and the connections between the units [4].

Although the symbolic approach allows very rich and expressive representations, it appears to have some intrinsic limitations such as the so-called 'symbol grounding problem,' [4] and the well known A.I. 'frame problem'.[5] On the other hand, the associationist approach suffers from its low-level nature, which makes it unsuited for complex tasks, and representations.

Gärdenfors' proposal of a third way of representing information exploits geometrical structures rather than symbols or connections between neurons. This geometrical representation is based on a number of what Gärdenfors calls 'quality dimensions' whose main function is to represent different qualities of objects such as brightness, temperature, height, width, depth.

Moreover, for Gärdenfors, judgments of similarity play a crucial role in cognitive processes. And, according to him, it is possible to associate the concept of distance to many kinds of quality dimensions. This idea naturally leads to the conjecture that the smaller is the distance between the representations of two given objects the more similar to each other the objects represented are.

[4] How to specify the meaning of symbols without an infinite regress deriving from the impossibility for formal systems to capture their semantics. See [6].

[5] Having to give a complete description of even a simple robot's world using axioms and rules to describe the result of different actions and their consequences leads to the 'combinatorial explosion' of the number of necessary axioms.

According to Gärdenfors, objects can be represented as points in a conceptual space, and concepts as regions within a conceptual space. These regions may have various shapes, although to some concepts—those which refer to natural kinds or natural properties[6]—correspond regions which are characterized by convexity.[7]

For Gärdenfors, this latter type of region is strictly related to the notion of prototype, i.e., to those entities that may be regarded as the archetypal representatives of a given category of objects (the centroids of the convex regions).

3 A non-phenomenological model of Conceptual Spaces

One of the most serious problems connected with Gärdenfors' conceptual spaces is that these have, for him, a phenomenological connotation. In other words, if, for example, we take, the conceptual space of colours this, according to Gärdenfors, must be able to represent the geometry of colour concepts in relation to how colours are given to us.

Now, since we believe that this type of approach is bound to come to grief as a consequence of the well-known problem connected with the subjectivity of the so-called '*qualia*', e.g., the specific and incommunicable quality of my visual perception of the rising Sun or of that ripe orange etc. etc., we have chosen a non phenomenological approach to conceptual spaces in which we substitute the expression 'measurement' for the expression 'perception', and consider a cognitive agent which interacts with the environment by means of the measurements taken by its sensors rather than a human being.

Of course, we are well aware of the controversial nature of our non phenomenological approach to conceptual spaces. But, since our main task in this paper is characterizing a rational agent with the view of providing a model for artificial agents, it follows that our non-phenomenological approach to conceptual spaces is justified independently of our opinions on qualia and their possible representations within conceptual spaces

Although the cognitive agent we have in mind is not a human being, the idea of simulating perception by means of measurement is not so far removed from biology. To see this, consider that human beings, and other animals, to survive need to have a fairly good ability to estimate distance. The frog unable to determine whether a fly is 'within reach' or not is, probably, not going to live a long and happy life.

Our CA is provided with sensors which are capable, within a certain interval of intensities, of registering different intensities of stimulation. For example, let us assume that CA has a visual perception of a green object h. If CA makes of the measure of the colour of h its present stereotype of green then it can, by means

[6] Actually, we do not agree with Gärdenfors when he asserts that:

> Properties... form a special case of concepts. [4], chapter 4, §4.1, p. 101.

[7] A set S is *convex* if and only if whenever $a, b \in S$ and c is between a and b then $c \in S$.

of a comparison of different measurements, introduce an ordering of gradations of green with respect to the stereotype; and, of course, it can also distinguish the colour of the stereotype from the colour of other red, blue, yellow, etc. objects. In other words, in this way CA is able to introduce a 'green dimension' into its colour space, a dimension within which the measure of the colour of the stereotype can be taken to perform the rôle of 0.

The formal model of a conceptual space that at this point immediately springs to mind is that of a metric space, i.e., it is that of a set X endowed with a metric. However, since the metric space X which is the candidate for being a model of a conceptual space has dimensions, dimensions the elements of which are associated with coordinates which are the outcomes of (possible) measurements made by CA, perhaps a better model of a conceptual space might be an n-dimensional vector space V over a field K like, for example, $\mathbb{R}^n$ (with the usual inner product and norm) on $\mathbb{R}$.

Although this suggestion is very interesting, we cannot help noticing that an important disanalogy between an n-dimensional vector space V over a field K, and the 'biological conceptual space' that V is supposed to model is that human, animal, and artificial sensors are strongly non-linear. In spite of its cogency, at this stage we are not going to dwell on this difficulty, because: (1) we intend to examine the 'ideal' case first; and because (2) we hypothesize that it is always possible to map a perceptual space into a conceptual space where linearity is preserved either by performing, for example, a small-signal approach, or by means of a projection onto a linear space, as it is performed in kernel systems [7].

4 Operating *in* and *on* Conceptual spaces

If our model of a conceptual space is, as we have repeatedly said, an n-dimensional vector space V over a field K, we need to distinguish between operating *in* V and operating *on* V. If we put $V = \mathbb{R}^n$ (over $\mathbb{R}$), then important examples of operations *in* $\mathbb{R}^n$ are the so-called 'rigid motions', i.e. all the functions from $\mathbb{R}^n$ into itself which are either real unitary linear functions[8] or translations.[9] Notice that if f is a rigid motion then f preserves distances, i. e. for any $v, w \in \mathbb{R}^n$, $d(v, w) = d(f(v), f(w))$. Examples of rigid motions which are real unitary linear functions are the θ-anticlockwise rotations of the x-axis in the x, y-plane.

To introduce operations *on* V, where V is an n-dimensional vector space over a field K, we need to make the following considerations. Let CA be provided with a set of measuring instruments which allow him to perform a finite set of measurements $M = \{m_1, \ldots, m_n\}$, and let $\{V_i\}_{i \in I}$ be the family of conceptual spaces— finite-dimensional vector spaces over a field K—present in CA's library.

[8] A linear function $f : \mathbb{R}^n \to \mathbb{R}^n$ is *real unitary* if and only if it preserves the inner product, i.e. for any $v, w \in \mathbb{R}^n$, we have $f(v) \cdot f(w) = v \cdot w$.

[9] The function $t : \mathbb{R}^n \to \mathbb{R}^n$ is a *translation* if and only if there exists a $v \in \mathbb{R}^n$ such that, for any $w \in \mathbb{R}^n$, we have $t(w) = w + v$.

If we assume that c is a point of one of these conceptual spaces, the coordinates $c_1, c_2, \ldots c_n$ of c represent particular instances of each quality dimension and, therefore, derive from the set of n measures performed by the agent on the subset of measurable elements. We, therefore, define two operations $\times$ and π *on* $\{V_i\}_{i \in I}$ such that: (1) $\times$ is the *direct product* of vector spaces, that is:

1. $V_i \times V_j = \{< v_i, v_j > \mid v_i \in V_i \text{ and } v_j \in V_j\}$;
2. for any $< v_{i,1}, v_{j,1} >, < v_{i,2}, v_{j,2} > \in V_i \times V_j$, we have: $< v_{i,1}, v_{j,1} > + < v_{i,2}, v_{j,2} > = < v_{i,1} + v_{i,2}, v_{j,1} + v_{j,2} >$
3. for any $k \in K$ and $< v_i, v_j > \in V_i \times V_j$, we have that: $k < v_i, v_j > = < kv_i, kv_j >$;

clearly, $V_i \times V_j$, for any $i, j \in I$, is a vector space, and

$$\dim (V_i \times V_j) = \dim V_i + \dim V_j;^{10}$$

and (2) π_i is the *projection* function onto the i-th coordinate space, i.e. $\pi_i(V_i \times V_j) = V_i$ and $\pi_j(V_i \times V_j) = V_j$, for $i, j \in I$. Obviously, we have that $\pi_i(V_i \times V_j)$ and $\pi_j(V_i \times V_j)$ are vector spaces, and that

$$dim\ \pi_i(V_i \times V_j) = dim\ V_i.$$

Now, with regard to the importance of the operator $\times$, consider that if we have the vector space $\mathbb{R}^3$, over the field $\mathbb{R}$, whose dimensions do not include time, we cannot then form the concept of velocity; and if the dimensions of the vector space $\mathbb{R}^3$, over the field $\mathbb{R}$, do not include colour, we cannot form the concept of red block. It is by producing, by means of $\times$, the right type of finite dimensional vector space that we make possible to formulate within it concepts such as velocity, red block, etc. The $\times$ operation on finite vector spaces has, to say it with Kant, an ampliative function. The relevance of π is, instead, all in its analytic rôle of explicating concepts by drawing attention to the elements belonging to a given coordinate space.

At each moment CA, instead of relying on the existence of a potentially infinite library of conceptual spaces, if necessary, individuates new dimensions following the procedure briefly illustrated on p. 3-4, and builds the current conceptual space suitable for the tasks that it has to accomplish by performing operations *on* the conceptual spaces which are already available.

5 A case study

We assume that CA is located on and can move around the floor of a room where objects of different type, size and color may be found. His sensors allow CA to obtain information concerning some of the characteristics of the surrounding environment and of some of the objects in it. When CA moves around the room, the perspective from which he views the objects present in the environment changes.

[10] $dim(V_i)$ is the dimension of the vector space V_i.

Of course, on the assumption that CA can tell from its receptors whether a given point of the floor of the room on which he is focussing is 'occupied' or not, it follows that CA is capable of performing tasks — like 'coasting around' the objects placed on the floor of the room — which do not require the use of conceptual spaces. But, on the other hand, there are tasks which require the use of systems of representation, such as conceptual spaces, which allow CA to build faithful representations (models) of the environment, etc.

Every time CA focuses its attention on something, CA identifies, *via* his receptors, the quality dimensions necessary for the representation of the object of interest and creates a specific current conceptual space individuating the regions (concepts) belonging to it.

To see this, assume that on the floor of the room where CA is there are two discs D_1 and D_2, and that CA's task consists in comparing in size D_1 with D_2. The initial current conceptual space V_0 of CA can be the vector space $\mathbb{R}^2$ (on $\mathbb{R}$) with the conceptual structure $\mathcal{C}_0$. CA is at the origin of the two axes of V_0 and the conceptual structure $\mathcal{C}_0$ associated to V_0 is $\mathcal{C}_0 = \{$FRONT (F), BACK (B), LEFT (L), RIGHT (R)$\}$. Here F, B, L, R are the *primitive* regions of V_0. (From now on, instead of talking about the conceptual space V_0 with structure $\mathcal{C}_0$, we shall simply consider the conceptual space $(V_0, \mathcal{C}_0)$.)

Note that the terms we use to refer to the *primitive* regions of V_0 are just a *façon de parler*, i.e., our way of describing the conceptual structure of the conceptual space of CA. In fact, we assume that the conceptual activity of CA is sub-linguistic.

CA can perform algebraic operations internal to the conceptual space which are mainly set operations given that the regions of V_0 are sets of points of V_0. The elementary operations defined on such regions are: $\cup, \cap, C_A^B$ (where $A \subseteq B$ and A and B are regions). Such operations applied to our primitive regions F, B, L, R allow us, for example, to individuate regions of particular importance such as the y-axis which can be characterized as the set of points $y \in C^{V_0}_{L \cup R}$, the x-axis as the set of points $x \in C^{V_0}_{F \cup B}$, the minimal region $\{0\}$, where 0 is the origin of the x and y axes as $C^{V_0}_{L \cup R} \cap C^{V_0}_{F \cup B} = \{0\}$, $F \cap R = \{(x, y) \mid 0 < x \text{ and } 0 < y\}$ (the first quadrant of $\mathbb{R}^2$), $L \cap R = \emptyset$, etc. As we have already seen at the very beginning of §3, another important class of operations internal to $(V_0, \mathcal{C}_0)$ are what we there called 'rigid motions'.

At this point we need to notice that $(V_0, \mathcal{C}_0)$ is a genuine conceptual space irrespective of the logic (first-order, second-order) used in studying it, because there is a difference between what CA does in constructing $(V_0, \mathcal{C}_0)$ and what the mathematician does in studying the properties of $(V_0, \mathcal{C}_0)$.

At the end of the exploration of the room on the part of CA, the current conceptual space will be $(V_1, \mathcal{C}_1)$, where V_1 is exactly like V_0 apart from the fact that a finite portion of it now models the room representing, for instance, within the conceptual structure of V_1 the sets of points corresponding to D_1 and D_2 by including within $\mathcal{C}_1$ the corresponding regions.

The task set to CA can now be accomplished within $(V_1, \mathcal{C}_1)$. In fact, CA can, without knowing what a circle, a disc, etc. are, translate D_1 onto D_2 and *vice versa*. (Remember that a translation is a rigid motion within $(V_1, \mathcal{C}_1)$.)

However, there is a task that CA cannot accomplish within a 2-d conceptual space, and this is: placing D_1 on top of D_2. To represent the situation CA needs a 3-d conceptual space, i.e., a vector space $X = \mathbb{R}^3$ (over $\mathbb{R}$) together with the appropriate conceptual structure $\mathcal{C}$. Of course, here X is obtained by means of the direct product of $\mathbb{R}^2$ by $\mathbb{R}$.

An interesting application of projection is the following which relates to a 3-d task that can be accomplished by means of a projection onto a 2-d conceptual space: seeing whether a given sphere lying on the floor fits into a cubic box placed next to it. Once again, our agent does not know what a sphere or a cube are, but can find a way of representing and solving the problem in a 2-d conceptual space by considering whether or not a maximum circle of the sphere can fit into a face of the cubic box.

6 Conclusions

In this paper we have introduced global operations which allow cognitive agents to build and rearrange their conceptual representations as a consequence of their perceptions and according to their goals.The proposed operations provide the agent with the capabilities to focus on and represent, in a proper current conceptual space, specific aspects of the perceived environment.

In order to evaluate the correctness of our proposal, we intend to produce a simulation environment within which to test on an artificial agent the efficiency of the model put forward

Acknowledgements

This work has been partially supported by the PON01_01687 - SINTESYS (Security and INTElligence SYSstem) Research Project.

References

1. A. Chella, M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. Artif. Intell., 89:73111, 1997.
2. Alessandra De Paola, Salvatore Gaglio, Giuseppe Lo Re, Marco Ortolani: An ambient intelligence architecture for extracting knowledge from distributed sensors. Int. Conf. Interaction Sciences 2009: 104-109.
3. HyunRyong Jung, Arjun Menon, Ronald C. Arkin. A Conceptual Space Architecture for Widely Heterogeneous Robotic Systems. Frontiers in Artificial Intelligence and Applications, Volume 233, 2011. Biologically Inspired Cognitive Architectures 2011, pp. 158 - 167. Edited by Alexei V. Samsonovich, Kamilla R. ISBN 978-1-60750-958-5
4. Gärdenfors, P.: 2004, Conceptual Spaces: The Geometry of Thought, MIT Press, Cambridge, Massachusetts.

5. Gardenfors, Peter (2004). Conceptual spaces as a framework for knowledge representation. Mind and Matter 2 (2):9-27.
6. S. Harnad. The symbol grounding problem. Physica D, 1990.
7. Bernhard Scholkopf and Alexander J. Smola. 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA.
8. Janet Aisbett and Greg Gibbon. 2001. A general formulation of conceptual spaces as a meso level representation. Artif. Intell. 133, 1-2 (December 2001), 189-232. DOI=10.1016/S0004-3702(01)00144-8 http://dx.doi.org/10.1016/S0004-3702(01)00144-8
9. Augello, A., Gaglio, S., Oliveri, G., Pilato, G. (2013). An Algebra for the Manipulation of Conceptual Spaces in Cognitive Agents. Biologically Inspired Cognitive Architectures, 6, 23-29.
10. Chella A., Frixione,M. Gaglio,S. (1998). An Architecture for Autonomous Agents Exploiting Conceptual Representations. Robotics and Autonomous Systems. Vol. 25, pp. 231?240 ISSN: 0921-8890.
11. Carsten Kessler, Martin Raubal - "Towards a Comprehensive Understanding of Context in Conceptual Spaces" - Workshop on Spatial Language in Context - Computational and Theoretical Approaches to Situation Specific Meaning. Workshop at Spatial Cognition, 19 September 2008
12. Parthemore, J. and A. Morse (2010). "Representations reclaimed: Accounting for the co-emergence of concepts and experience", Pragmatics and Cognition, 18 (2), pp. 273-312.

Using relational adjectives for extracting hyponyms from medical texts

Olga Acosta[α], César Aguilar[α] & Gerardo Sierra[βγ]

[α]Department of Language Sciences, Pontificia Universidad Católica de Chile
[β]Engineering Institute, Universidad Nacional Autónoma de México, Mexico
[γ]Universite d'Avignon et des Pays de Vaucluse, France
olgalimx@gmail.com
caguilara@uc.cl/http://cesaraguilar.weebly.com
gsierram@iingen.unam.mx/www.iling.unam.mx

Abstract. We expose a method for extracting hyponyms and hypernyms from analytical definitions, focusing on the relation observed between hypernyms and relational adjectives (e.g., *cardiovascular disease*). These adjectives introduce a set of specialized features according to a categorization proper to a particular knowledge domain. For detecting these sequences of hypernyms associated to relational adjectives, we perform a set of linguistic heuristics for recognizing such adjectives from others (e.g. *psychological/ugly disorder*). In our case, we applied linguistic heuristics for identifying such sequences from medical texts in Spanish. The use of these heuristics allows a trade-off between precision & recall, which is an important advance that complements other works.

Keywords: Hypernym/hyponym, lexical relation, analytical definition, categorization, prototype theory.

1 Introduction

One relevant line of research into NLP is the automatic recognition of lexical relations, particularly hyponymy/hyperonymy (Hearts 1992; Ryu and Choy 2005; Pantel and Pennacchiotti 2006; Ritter, Soderland, and Etzioni 2009). In Spanish Acosta, Aguilar and Sierra (2010); Ortega et al. (2011); and Acosta, Sierra and Aguilar (2011) have reported good results detecting hyponymy/hyperonymy relations in corpus of general language, as well as specialized corpus on medicine.

From a cognitive point of view, hyponymy/hyperonymy lexical relation is a process of *categorization*, which implies that these relations allow recognizing, differentiating and understanding entities according to a set of specific features. Following the works of Rosch (1978), Smith and Medin (1981), as well Evans and Green (2006), hypernyms are associated to *basic levels* of categorization. If we considered a taxonomy, the *basic level* is a level where categories carry the most information, as well they possess the highest cue validity, and are the most differentiated from one another (Rosch, 1978). In other words, as Murphy (2002) points out, *basic level* (e.g., *chair*) can represent a compromise between the accuracy of classification at a higher superordinate category (e.g., *furniture*) and the predictive power of a subordinate category (e.g., *rocking chair*). However, as Tanaka and Taylor's (1991) study showed, in spe-

cific domains experts primarily use subordinate levels because of they know more distinctive features of their entities than novices do. In this work, we propose a method for extracting these subordinate categories from hypernyms found in analytical definitions.

We develop here a method for extracting hyponymy-hyper**o**nymy relations from analytical definitions in Spanish, having in mind this process of categorization. We perform this extraction using a set of syntactic patterns that introduce definitions on texts. Once we obtained a set of candidates to analytical definitions, we filter this set considering the most common hyperonyms (in this case, the Genus terms of such definitions), which are detected by establishing specific frequency thresholds. Finally, the most frequent hypernym subset is used for extracting subordinate categories. We prioritize here relational adjectives because they associate a set of specialized properties to a noun (that is, the hypernym).

2 Concept theories

Categorization is one of the most basic and important cognitive processes. Categorization involves recognizing a new entity as part of abstract something conceived with other real instances (Croft and Cruse, 2004). Concepts and categories are two elements that cannot be seen separated each other. As Smith and Medin (1981) point out, concepts have a categorization function used for classifying new entities and extracting inferences about them.

Several theories have been proposed in order to explain formation of concepts. The classical theory (Aristotelian) holds that all instances of a concept share common properties, and that these common properties are necessary and sufficient to define the concept. However, classical approach did not provide explanation about many concepts, This fact led to Rosch to propose the prototype theory (1978) which explains, unlike to the classical theory, the instances of a concept differ in the degree to which they share certain properties, and consequently show a variation respect to the degree of representation of such concept. Thus, prototype theory provides a new view in which a unitary description of concepts remains, but where the properties are true of most, and not all members. On the other hand, exemplar theory holds that there is no single representation of an entire class or concept; categories are represented by specific exemplars instead of abstracted prototypes (Minda and Smith, 2002).

Finally, as mentioned in section 1, prototype theory supports existence of a hierarchical category system where a basic level is the most used level. In this work we assumed this basic level is genus found in analytical definitions, so that we use it for extracting subordinate categories.

2.1 Principles of categorization

Rosch (1978) proposes two principles in order to build a system of categories. The first refers to the function of this system, which must provide a maximum of information with the least cognitive effort. The second emphasizes that perceived world (not-metaphysical) has structure. Maximum information with least cognitive effort is

achieved if categories reflect the structure of the perceived world as better as possible. Both the cognitive economy principle and the structure of perceived world have important implications in the construction of a system of categories.

Rosch conceives two dimensions in this system: vertical and horizontal. Vertical dimension refers to the category's level of inclusiveness, that is, the subsumption relation between different categories. In this sense, each subcategory C' must be a proper subset from its immediately preceding category C, that is:

$$C' \subset C, \text{ where } |C'| < |C| \tag{1}$$

The implications of both principles in the vertical dimension are that not all the levels of categorization C are equally useful. There are basic and inclusive levels c_i^b where categories can reflect the structure of attributes perceived in the world. This inclusiveness level is the mid-part between the most and least inclusive levels, that is:

$$c_i^b \subset c_j^{\sup} \; and \; c_k^{sub} \subset c_i^b \text{ , for i, j, k} > 0 \tag{2}$$

In the figure 1, basic levels c_i^b are associated with categories such as *car*, *dog* and *chair*. Categories situated on the top of the vertical axis —which provide less detail— are called superordinate categories c_j^{sup} (*vehicle*, *mammal*, and *furniture*). In contrast, those located in the lower vertical axis, which provide more detail, are called subordinate categories c_k^{sub} (*saloon*, *collie*, and *rocking chair*).

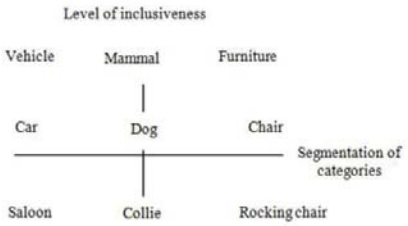

Fig. 1. The human categorization system (extracted from Evans and Green 2006)

On the other hand, horizontal dimension focuses on segmentation of categories in the same level of inclusiveness, that is:

$$\bigcup_{i=1}^{n} C_i = C \text{ , where } C_i \cap C_k = \varnothing, \; i \neq k \tag{3}$$

Where n represents number of subcategories C_i within category C. Ideally, these subcategories must be a relevant partition from C. The implications of these principles of categorization in the horizontal dimension are that —when there is an increase in the level of differentiation and flexibility of the categories C_i— they tend to be defined in

terms of prototypes. These prototypes have the most representative attributes of instances within a category, and fewer representative attributes of elements of others. This horizontal dimension is related to the principle of structure of the perceived world.

2.2 Levels of categorization

Studies on cognitive psychology reveal the prevalence of basic levels in natural language. Firstly, basic level terms tend to be monolexemic (*dog*, *car*, *chair*); in contrast, subordinate terms have at least two lexemes (e.g.: *rocking chair*), and often include basic level terms (Murphy 2002; Minda and Smith 2002, Croft and Cruse 2004; Evans and Green 2006). Secondly, the basic level is the most inclusive and the least specific for delineating a mental image. Thus, if we considered a superordinate level, it is difficult to create an image of the category, e.g.: *furniture*, without thinking in a specific item like a *chair* or a *table*. Despite preponderance of the basic level, superordinate and subordinate levels also have very relevant functions. According to Croft and Cruse (2004), superordinate level emphasizes functional attributes of the category, and also performing a collecting function. Meanwhile, subordinate categories achieve a function of specificity. Given the function of specificity of subordinate categories in specialized domains, we consider them are important for building lexicons and taxonomies.

3 Subordinate categories of interest

Let *H* be set of all single-word hyperonyms implicit in a corpus, and *F* the set of the most frequent hyperonyms in a set of candidate analytical definitions by establishing a specific frequency threshold m:

$$F = \{x \mid x \in H, \text{freq}(x) \geq m\} \quad (4)$$

On the other hand, *NP* is the set of noun phrases representing candidate categories:

$$NP = \{np \mid \text{head}(np) \in F, \text{modifier}(np) \in \text{adjective}\} \quad (5)$$

Subordinate categories *C* of a basic level b are those holding:

$$C^b = \{np \mid \text{head}(np) \in F, \text{modifier}(np) \in \text{relational-adjective}\} \quad (6)$$

Where modifier (*np*) represents an adjective inserted on a noun phrase *np* with head *b*. We hope these subcategories reveal important division perspectives of a basic level. In this work we only focused on relational adjectives, although prepositional phrases can generate relevant subordinate categories (e.g., *disease of Lyme* or *Lyme disease*).

4 Types of adjectives

According to Demonte (1999), adjectives are a grammatical category whose function is to modify nouns. There are two kinds of adjectives which assign properties to nouns: attributive and relational adjectives. On the one hand, descriptive adjectives refer to constitutive features of the modified noun. These features are exhibited or characterized by means of a single physical property: color, form, character, predisposition, sound, etc.: *el libro azul* (the blue book), *la señora delgada* (the slim lady). On the other hand, relational adjectives assign a set of properties, e.g., all of the characteristics jointly defining names as: *puerto marítimo* (maritime port), *paseo campestre* (country walk). In terminological extraction, relational adjectives represent an important element for building specialized terms, e.g.: *inguinal hernia*, *venereal disease, psychological disorder* and others are considered terms in medicine. In contrast, *rare hernia*, *serious disease* and *critical disorder* seem more descriptive judgments.

5 Methodology

We expose here our methodology for extracting first conceptual information, and then recognizing our candidates of hyponyms.

5.1 Automatic extraction of analytical definitions

We assume that the best sources for finding hyponymy-hyperonymy relations are the definitions expressed in specialized texts, following to Sager and Ndi-Kimbi (1995), Pearson (1998), Meyer (2001), as well Klavans and Muresan (2001). In order to achieve this goal, we take into account the approach proposed by Acosta et al. (2011). Figure 2 shows an overview of the general methodology, where input is a non-structured text source. This text source is tokenized in sentences, annotated with POS tags and normalized. Then, syntactical and semantic filters provide the first candidate set of analytical definitions. Syntactical filter consists on a chunk grammar considering verb characteristics of analytical definitions, and its contextual patterns (Sierra *et al*., 2008), as well as syntactical structure of the most common constituents such as term, synonyms, and hyperonyms. On the other hand, semantic phase filters candidates by means of a list of noun heads indicating relations part-whole and causal as well as empty heads semantically not related with term defined. An additional step extracts terms and hyperonyms from candidate set.

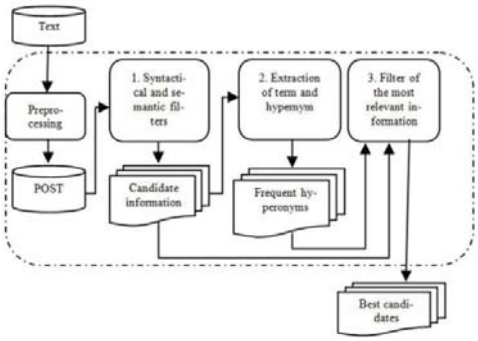

Fig. 2. Methodology for extracting analytical definitions

5.2 Extraction of subordinate categories

As in the case of terms, we consider relational adjectives and prepositional phrases are used for building subordinate categories in specialized domains, but in this work we only focused on relational adjectives. Thus, we use the most frequent hyperonyms for extracting these relevant subordinate categories. In first place, we obtain a set of noun phrases with structure: noun + adjective from corpus, as well as its frequency. Then, noun phrases with hyperonyms as head are selected, and we calculate the pointwise mutual information (PMI) for each combination. Given its use in collocation extraction, we select a PMI measure, where PMI thresholds are established in order to filter non-relevant (NR) information. We considered the normalized PMI measure proposed by Bouma (2009):

$$i_n(x,y) = \left(\ln \frac{p(x,y)}{p(x)p(y)} \right) \Big/ -\ln p(x,y) \tag{7}$$

This normalized variant is due to two fundamental issues: to use association measures whose values have a fixed interpretation, and to reduce sensibility to low frequencies of data occurrence.

6 Results

In these sections we expose the results of our experiments.

6.1 Text source

Our source is a set of medical documents, basically human body diseases and related topics (surgery, treatments, and so on). These documents were collected from *MedLinePlus* in Spanish. *MedLinePlus* is a site whose goal is to provide information about diseases and conditions in an accessible way of reading. The size of the corpus is 1.3 million of words. We chose a medical domain for reasons of availability of textual resources in digital format. Further, we assume that the choice of this domain does not suppose a very strong constraint for generalization of results to other domains.

6.2 Programming language and tools

Programming language used for automatizing all of the tasks was Python and NLTK module (Bird, Klein and Loper 2009). Our proposal is based on lexical-syntactical patterns, so that we assumed as input a corpus with POS tags. POS tagged was done with TreeTagger (Schmid 1994).

6.3 Some problems for analyzing

In these sections we delineate some important problems detected in our experiment: the recognition to a relation of semantic compositionality between hyperonyms.

6.3.1 Semantic compositionality between hyperonyms and relational adjectives

We understand semantic compositionality as a regulation principle that assigns a specific meaning to each of lexical units in a phrase structure, depending on the syntactical configuration assuming such structure (Partee, 1995). Specific combinations of lexical units determine the global meaning of a phrase or sentence generating not only isolated lexical units, but blocks which refer to specific concepts (Jackendoff, 2002). Given this principle, a term as *gastrointestinal inflammation* operates as a hyponym or subordinate category with more wealth of specific information, than the hypernym *inflammation*.

6.3.2 Hypernym and its lexical fields

Hypernyms, as generic classes of a domain, are expected to be related to a great deal of modifiers such as adjectives, nouns and prepositional phrases reflecting more specific categories (e.g., *cardiovascular disease*) than hyperonyms, or simply sensitive descriptions to a specific context (e.g., *rare disease*). As an illustrative example and only for the case of adjective modifiers, table 1 shows the disease hypernym and the first most related subset of 50 adjectives, taking into account its PMI values. In this example extracted of a real corpus, only 30 out of 50 (60%) are relevant relations. In total, disease is related to 132 adjectives, of which, 76 (58%) can be considered relevant.

Table 1. The first 50 adjectives with most high PMI value

C(enfermedad, w_i)
Transmisible, prevenible, diarreica, diverticular, indicadora, autoinmunitaria, aterosclerótica, meningocócica, cardiovascular, pulmonar, afecto, febril, agravante, hepática, seudogripal, periodontal, sujeto, bacteriano, emergente, benigno, parasitaria, postrombótica, bacteriémica, coexistente, catastrófica, exclusiva, vectorial, supurativa, infecciosa, debilitante, digestiva, invasora, rara, inflamatoria, esporádica, antimembrana, predisponente, ulcerosa, contagiosa, cardiaca, sistémica, activa, grave, prexistente, miocárdica, somática, fulminante, atribuible, linfoproliferativa

On the other hand, if we consider a relational adjective, for example, cardiovascular, we find that it modifies to a set of nouns, as shown in table 2. The case of a descriptive adjective as rare is similar; it also modifies a set of nouns. Thus, we have both relational and descriptive adjectives can be linked with other elements, this situation mirrors how the compositionality principle operates, decreasing precision to the association measures for detecting relevant relations.

Table 2. Nouns modified by relational adjective cardiovascular and descriptive adjective rare

C(w_i, cardiovascular)	C(w_i, raro)
efecto, problema, congreso, función, evento, relación, examen, inestabilidad, trastorno, enfermedad, bypass, causa, beneficio, sistema, reparador, descompensación, cirugía, operación, mortalidad, aparato, educación, síntoma, eficiencia, episodio, riesgo, investigación, manifestación, afección, medicamento, director, muerte, salud	televisión, enfermedad, complicación, infancia, niño, color, obesidad, mhc, nucleótido, sustancia, mutación, trastorno, grupo, meconio, epistaxis, derecha, síndrome, cáncer, alelo, forma, caso, párpado

6.3.3 Linguistic heuristics for filtering non-relevant adjectives

In order to face the phenomenon of compositionality between hyperonyms and relational adjectives that affect the performance of traditional measures, we automatically extract a stop-list of descriptive adjectives from the same source of input information, implementing three criteria proposed in Demonte (1999) for distinguishing between descriptive and relational adjectives. These criteria are:

- Adjective used predicatively: *The method is important*.
- Adjective used in comparisons, so that its meaning is modified by adverbs of degree: *relatively fast*.
- Precedence of adjective respect to the noun: *A serious disease*.

6.4 Automatic extraction of conceptual information

We consider two approaches based on patterns, and a baseline derived from only most common verbs used in analytical definitions. Both of the methods outperformed baseline's precision, but recall was significantly decreased. On the one hand, the method proposed by Sierra et al. (2008) achieved a good recall (63%), but the precision was very low (24%). On the other hand, with the method proposed by Acosta et al. (2011) we achieved a high precision (68%), and a trade-off between precision and recall (56%). Given that this latter method achieved the better results, we decided to implement it in order to obtain our set of hyperonyms necessary for the next phase of extraction of subordinate categories.

Table 3. Extraction of analytical definitions

	Precision	Recall	F-Measure
Baseline	8%	100%	15%
Sierra *et al.* (2008)	24%	63%	35%
Acosta *et al.* (2011)	68%	56%	61%

6.5 Extraction to subordinate categories

We extract a set of descriptive adjectives by implementing linguistic heuristics. Our results show a high precision (68%) with a recall acceptable (45%). This subset of descriptive adjectives is removed from the set of noun phrases with structure: noun + adjective before final results. Table 4 shows the initial precision, that is, precision obtained without some filtering process.

Table 4. Initial precision

Hyperonym	Initial precision	Hyperonym	Initial precision
Enfermedad	61	Tratamiento	36
Desorden	80	Cirugía	67
Examinación	52	Método	37
Condición	61	Problema	62
Procedimiento	40	Proceso	47
Infección	56	Inflamación	58
Proteína	67	Glándula	95
Cáncer	55	Órgano	43
Tumor	63	Medicamento	60

This precision is compared with precision by setting several PMI thresholds (0, 0.10, 0.15, and 0.25) as shown in table 5. Results show a significant improvement in precision from PMI 0.25, but recall is negatively affected as this threshold is increased. On the other hand, if we consider linguistic heuristics we obtain a trade-off between precision and recall, as shown in table 6.

7 Final considerations

In this paper we present a comparison between two approaches for automatically extracting subordinate categories arising from a hypernym within a domain of medical knowledge.

The main point in this discussion is the possibility to generate a lot of relevant hyponyms having as head a hypernym. Unfortunately, given the generic nature of the single-word hypernyms, these can be directly linked with a large amount of modifiers such as nouns, adjectives and prepositional phrase, so that to extract the most relevant subordinate categories with traditional measures become a very complex task.

In this paper we only consider relational adjectives, because we consider they are best candidates for codifying subordinate categories. It is remarkable the high degree of compositionality present in the relation between hyperonyms and relational adjectives, which is detrimental to the accuracy of measures of association to select relevant relations. It is just in these scenarios where the regularity of language, according to Manning and Schütze (1999) acquires great importance for assisting methods such as parsing, lexical/semantic disambiguation and, in our particular case, extracting relevant hyponyms.

Table 5. Precision (P), recall (R) and F-Measure (F) by PMI threshold

Hyperonym	PMI >= 0			PMI >= 0.10			PMI >= 0.15			PMI >= 0.25		
	P	R	F	P	R	F	P	R	F	P	R	F
Enfermedad	66	89	76	69	74	71	68	61	64	74	30	43
Desorden	82	92	87	85	79	82	85	74	79	87	58	70
Examinación	55	96	70	62	85	72	67	78	72	76	69	72
Condición	62	97	76	62	86	72	66	78	72	69	55	61
Procedimiento	40	100	57	43	100	60	40	84	54	57	68	62
Infección	58	92	71	64	81	72	67	67	67	73	47	57
Proteína	67	100	80	71	100	83	74	100	85	83	89	86
Cáncer	59	97	73	56	81	66	57	72	64	61	61	61
Tumor	63	100	77	64	88	74	66	83	74	69	64	66
Tratamiento	36	90	51	39	78	52	45	73	56	52	47	49
Cirugía	68	100	81	73	91	81	74	84	79	72	52	60
Método	37	100	54	39	100	56	39	93	55	38	64	48
Problema	64	93	76	63	75	68	65	54	59	67	29	40
Proceso	47	100	64	50	95	66	50	86	63	53	65	58
Inflamación	57	94	71	55	89	68	57	89	69	52	61	56
Glándula	95	100	97	95	100	97	95	100	97	95	95	95
Órgano	44	100	61	40	82	54	39	71	50	43	71	54
Medicamento	60	98	74	69	94	80	67	92	78	77	81	79

Table 6. Precision, recall and F-measure by linguistic heuristics

Hyperonym	Linguistic Heuristics		
	P	R	F
Enfermedad	79	74	76
Desorden	95	69	80
Examinación	74	85	79
Condición	88	75	81
Procedimiento	85	89	87
Infección	84	76	82
Proteína	88	76	82
Cáncer	69	94	80
Tumor	82	86	84
Tratamiento	68	70	69
Cirugía	90	82	86
Método	72	82	77
Problema	83	67	74
Proceso	73	73	73
Inflamación	82	78	80
Glándula	100	100	100
Órgano	71	71	71
Medicamento	76	72	74

8 Aknowledgements

We would like to acknowledge the sponsorship of the project CONACYT CB2012/178248 “Detección y medición automática de similitud textual”.

References

1. Acosta, O., C. Aguilar, and G. Sierra. 2010. A Method for Extracting Hyponymy-Hypernymy Relations from Specialized Corpora Using Genus Terms. In Proceedings of the Workshop in Natural Language Processing and Web-based Technologies 2010, 1-10, Córdoba, Argentina, Universidad Nacional de Córdoba.
2. Acosta, O., G. Sierra, and C. Aguilar. 2011. Extraction of Definitional Contexts using Lexical Relations. International Journal of Computer Applications, 34(6): 46-53.
3. Bird, S., Klein, E., and Loper. E. 2009. Natural Language Processing whit Python. O'Reilly. Sebastropol, Cal.
4. Bouma, G. 2009. Normalized (Pointwise) Mutual Information in Collocation Extraction. In From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference, 31-40, Gunter Narr Verlag, Tübingen, Germany.
5. Croft, W., and D. Cruse. 2004. Cognitive Linguistics. Cambridge University Press, Cambridge, UK.
6. Demonte, V. 1999. El adjetivo. Clases y usos. La posición del adjetivo en el sintagma nominal. In Gramática descriptiva de la lengua española, Vol. 1, Chapter. 3: 129-215, Espasa-Calpe Madrid, Spain.

7. Evans, V., and Green, M. 2006. Cognitive Linguistics: An Introduction. LEA, Hillsdale, New Jersey.
8. Hearst, M. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In Proceedings of the Fourteenth International Conference on Computational Linguistics, 539-545, Nantes, France.
9. Jackendoff, R. 2002. Foundations of Language: Brain, Meaning, Grammar, Evolution, Oxford University Press, Oxford, UK.
10. Klavans, J. and Muresan, S. 2001. Evaluation of the DEFINDER system for fully automatic glossary construction. In Proceedings of the American Medical Informatics Association Symposium, 252-262, ACM Press, New York.
11. Manning, Ch., and Schütze, H. 1999. Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, Mass.
12. Meyer, I. 2001. Extracting knowledge-rich contexts for terminography. In Bourigault, D., Jacquemin, C. and L'Homme, M.C. (eds.). Recent Advances in Computational Terminology, 127-148, John Benjamins, Amsterdam/Philadelphia.
13. Minda, J., and Smith, J. 2002. Comparing Prototype-Based and Exemplar-Based Accounts of Category Learning and Attentional Allocation. Journal of Experimental Psychology 28(2): 275–292.
14. Murphy, G. 2002. The Big Book of Concepts, MIT Press, Cambridge, Mass.
15. Ortega, R., C. Aguilar, L. Villaseñor, M. Montes and G. Sierra. 2011. Hacia la identificación de relaciones de hiponimia/hiperonimia en Internet. Revista Signos 44(75): 68-84.
16. Pantel, P. & Pennacchiotti, M. 2006. Espresso: Lever-aging Generic Patterns for Automatically Harvesting Semantic Relations. In 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 113-120, Sydney, Australia.
17. Partee, B. 1995. Lexical Semantics and Compositionality. In Invitation to Cognitive Science, Part I: Language, 311-336, MIT Press, Cambridge, Mass.
18. Pearson, J. 1998. Terms in Context. John Benjamins, Amsterdam/Philadelphia.
19. Ritter, A., Soderland, S., and Etzioni, O. 2009. What is This, Anyway: Automatic Hypernym Discovery. In Papers from the AAAI Spring Symposium, 88-93. Menlo Park, Cal.: AAAI Press.
20. Rosch, E. 1978. Principles of categorization. In Rosh, E. and Lloyd, B. (eds.), Cognition and Categorization, Chapter 2, 27-48. LEA, Hillsdale, New Jersey.
21. Ryu, K., and Choy, P. 2005. An Information-Theoretic Approach to Taxonomy Extraction for Ontology Learning. In Buitelaar, P., Cimiano, P., and Magnini, B. (eds.) Ontology Learning from Text: Methods, Evaluation and Applications, 15-28. IOS Press, Amsterdam.
22. Sager, J. C., and Ndi-Kimbi, A. 1995. The conceptual structure of terminological definition and their linguistic realisations: A report on research in progress. Terminology 2(1): 61-85.
23. Sierra, G., Alarcón, R., Aguilar, C., and Bach, C. 2008. Definitional verbal patterns for semantic relation extraction", Terminology 14(1): 74-98.
24. Schmid, H. 1994. Probabilistic Part-of-Speech Tag-ging Using Decision Trees. In Proceedings of In-ternational Conference of New Methods in Language. WEB Site: www.ims.uni-stuttgart.de~schmid.TreeTagger.
25. Smith, E., and Medin, D. 1981. Categories and Concepts, Cambridge, Mass.: Harvard University Press.
26. Tanaka, J., and Taylor, M. 1991. Object categories and expertise: Is the basic level in the eye of the beholder? Cognitive Psychology, 15, 121–149.

Controlling a General Purpose Service Robot By Means Of a Cognitive Architecture

Jordi-Ysard Puigbo[1], Albert Pumarola[1], and Ricardo Tellez[2]

[1] Technical University of Catalonia
[2] Pal Robotics ricardo.tellez@pal-robotics.com

Abstract. In this paper, a humanoid service robot is equipped with a set of simple action skills including navigating, grasping, recognizing objects or people, among others. By using those skills the robot has to complete a voice command in natural language that encodes a complex task (defined as the concatenation of several of those basic skills). To decide which of those skills should be activated and in which sequence no traditional planner has been used. Instead, the SOAR cognitive architecture acts as the reasoner that selects the current action the robot must do, moving it towards the goal. We tested it on a human size humanoid robot Reem acting as a general purpose service robot. The architecture allows to include new goals by just adding new skills (without having to encode new plans).

1 Introduction

Service robotics is an emerging application area for human-centered technologies. Even if there are several specific applications for those robots, a general purpose robot control is still missing, specially in the field of humanoid service robots [1]. The idea behind this paper is to provide a control architecture that allows service robots to generate and execute their own plan to accomplish a goal. The goal should be decomposable into several steps, each step involving a one step skill implemented in the robot. Furthermore, we want a system that can openly be increased in goals by just adding new skills, without having to encode new plans.

Typical approaches to general control of service robots are mainly based on state machine technology, where all the steps required to accomplish the goal are specified and known by the robot before hand. In those controllers, the list of possible actions that the robot can do is exhaustively created, as well as all the steps required to achieve the goal. The problem with this approach is that everything has to be specified beforehand, preventing the robot to react to novel situations or new goals.

An alternative to state machines is the use of planners [2]. Planners decide at running time which is the best sequence of skills to be used in order to achieve the goal specified, usually based on probabilistic approaches. A different approach to planners is the use of cognitive architectures. Those are control systems that

try to mimic some of the processes of the brain in order to generate a decision [3][4][5][6][7][8].

There are several cognitive architectures available: SOAR [9], ACT-R [10, 11], CRAM [12], SS-RICS [5], [13]. From all of them, only CRAM has been designed with direct application to robotics in mind, having been applied to the generation of pan cakes by two service robots [14]. Recently SOAR has also been applied to simple tasks of navigation on a simple wheeled robot [15].

At time of creating this general purpose service robot, CRAM was only able to build plans defined beforehand, that is, CRAM is unable to solve unspecified (novel) situations. This limited the actions the robot could do to the ones that CRAM had already encoded in itself. Because of that, in our approach we have used the SOAR architecture to control a human sized humanoid robot Reem equipped with a set of predefined basic skills. SOAR selects the required skill for the current situation and goal, without having a predefined list of plans or situations.

The paper is structured as follows: in section 2 we describe the implemented architecture, in section 3, the robot platform used. Section 4 presents the results obtained and we end the paper with the conclusions.

2 Implementation

The system is divided into four main modules that are connected to each other as shown in the figure 1. First, the robot listens a vocal command and translates it to text using the automatic speech recognition system (ASR). Then, the semantic extractor divides the received text into grammatical structures and generates a goal with them. In the reasoner module, the goal is compiled and sent to the cognitive architecture (SOAR). All the actions generated by SOAR are translated into skill activations. The required skill is activated through the action nodes.

2.1 Automatic Speech Recognition

In order to allow natural voice communication the system incorporates a speech recognition system capable of processing the speech signal and returns it as text for subsequent semantic analysis. This admits a much natural way of Human-Robot Interaction (HRI). The ASR is the system that allows translation of voice commands into written sentences.

The ASR software used is based on the open source infrastructure Sphinx developed by Carnegie Mellon University [16]. We use a dictionary that contains 200 words which the robot understands. In case the robot receives a command with a non-known word the robot will not accept the command and is going to request for a new command.

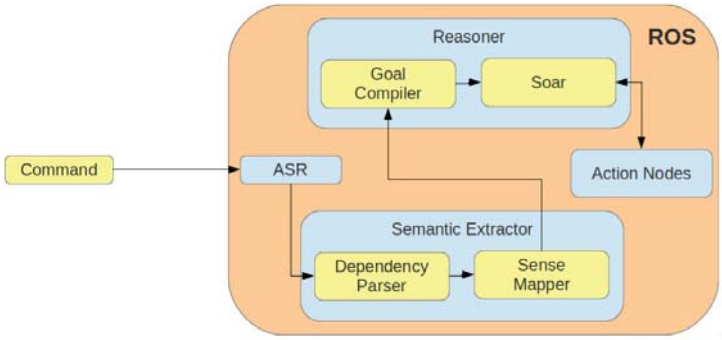

Fig. 1. Diagram of the system developed

2.2 Semantic Extractor

The semantic extractor is the system in charge of processing the imperative sentences received from the ASR, extracting and retrieving the relevant knowledge from it.

The robot can be commanded using two types of sentences:

Category I The command is composed by one or more short, simple and specific subcommands, each one referring to very concrete action.

Category II The command is under-specified and requires further information from the user. The command can have missing information or be composed of categories of words instead of specific objects (ex. *bring me a coke* or *bring me a drink*. First example does not include information about where the drink is. Second example does not explain which kind of drink the user is asking for).

The semantic extractor implemented is capable of extracting the subcommands contained on the command, if these actions are connected in a single sentence by conjunctions (*and*), transition particles (*then*) or punctuation marks. Should be noticed that, given that the output comes from an ASR software, all punctuation marks are omitted.

We know that a command is commonly represented by an imperative sentence. This denotes explicitly the desire of the speaker that the robot performs a certain action. This action is always represented by a verb. Although a verb may convey an occurrence or a state of being, as in *become* or *exist*, in the case of imperative sentences or commands the verb must be an action. Knowing this, we asume that any command will ask the robot to do something and these actions might be performed involving a certain object (*grasp a coke*), location (*navigate to the kitchen table*) or a person (*bring me a drink*). In *category I* commands, the semantic extractor should provide the specific robot action and the object, location or person that this action has to act upon. In *category II*, commands do

not contain all the necessary information to be executed. The semantic extractor must figure out which is the action, and identify which information is missing in order to accomplish it.

For semmantic extraction we constructed a parser using the Natural Language ToolKit (NLTK) [17]. A context-free grammar (CFG) was designed to perform the parsing. Other state-of-the-art parsers like Stanford Parser [18] or Malt Parser [19] were discarded for not having support for imperative sentences, having been trained with deviated data or needing to be trained beforehand. It analyses dependencies, prepositional relations, synonyms and, finally, co-references.

Using the CFG, the knowledge retrieved from each command by the parser is stored on a structure called parsed-command. It contains the following information:

- Which action is needed to perform
- Which location is relevant for the given action
- Which object is relevant for the given action
- Which person is relevant for the given action

The parsed-command is enough to define most goals for a service robot at home, like *grasp - coke* or *bring - me - coke*. For multiple goals (like in the category I sentences), an array of parsed-commands is generated, each one populated with its associated information.

The process works as follows: first the sentence received from the ASR is tokenized. Then, NLTK toolkit and Stanford Dependency Parser include already trained Part-Of-Speech (POS) tagging functions for English. Those functions complement all the previous tokens with tags that describe which is the POS more plausible for each word. By applying POS-tagging, the verbs are found. Then, the action field of the parsed-command is filled with the verb.

At this point the action or actions that are needed to eventually accomplish the command have been already extracted. Next step is to obtain their complements. To achieve this a combination of two methods is used:

1. Identifying from all the nouns in a sentence, which words are objects, persons or locations, using an ontology.
2. Finding the dependencies between the words in the sentence. Having a dependency tree allows identification of which parts of the sentence are connected to each other and, in that case, identify which connectors do they have. This means that finding a dependency tree (like for example, the Stanford Parser), allows to find which noun acts as a direct object of a verb. Additionally, looking for the direct object, allows us to find the item over which the action should be directed. The same happens with the indirect object or even locative adverbials.

Once finished this step, the full parsed-command is completed. This structure is sent to the next module, where it will be compiled into a goal interpretable by the reasoner.

2.3 Reasoner

Goal Compiler A compiler has been designed to produce the goal in a format understandable by SOAR from the received parsed-command, called the compiled-goal.

It may happen that the command lacks some of the relevant information to accomplish the goal (*category II*). This module is responsible for asking the questions required to complete this missing information. For example, in the command "*bring me a drink*", knowing that a *drink* is a category, the robot will ask for which drink is asking the speaker. Once the goals are compiled they are sent to SOAR module.

SOAR SOAR module is in charge of deciding which skills must be executed in order to achieve the compiled-goal. A loop inside SOAR selects the skill that will move Reem one step closer to the goal. Each time a skill is selected, a petition is sent to an action node to execute the corresponding action. Each time a skill is executed and finished, SOAR selects a new one. SOAR will keep selecting skills until the goal is accomplished.

The set of skills that the robot can activate are encoded as operators. This means that there is, for each possible action:

- A rule proposing the operator, with the corresponding name and attributes.
- A rule that sends the command through the output-link if the operator is accepted.
- One or several rules that depending on the command response, fire and generate the necessary changes in the world.

Given the nature of the SOAR architecture, all the proposals will be treated at the same time and will be compared in terms of preferences. If one is best than the others, this one is the only operator that will execute and a new deliberation phase will begin with all the new available data. It's important to know that all the rules that match the conditions are treated as if they fired at the same time, in parallel. There is no sequential order [20].

Once the goal or list of goals have been sent to SOAR the world representation is created. The world contains a list of robots, and a list of objects, persons and locations. Notice that, at least, there is always one robot represented, the one that has received the command, but, instead of just having one robot, one can generate a list of robots and because of the nature of the system they will perform as a team of physical agents to achieve the current goal.

SOAR requires an updated world state, in order to make the next decision. The state is updated after each skill execution, in order to reflect the robot interactions with the world. The world could be changed by the robot itself or other existing agents. Changes in the world made by the robot actions directly reflect the result of the skill execution in the robot world view. Changes in the world made by other agents, may make the robot fail the execution of the current skill, provoking the execution of another skill that tries to solve the impasse (for

example, going to the place where the coke is and finding that the coke is not there any more, will trigger the *search for object* skill to figure out where the coke is).

This means that after the action resolves, it returns to SOAR an object describing the success/failure of the action and the relevant changes it provoked. This information is used to change the current knowledge of the robot. For instance, if the robot detected a beer bottle and its next skill is to grasp it, it will send the command 'grasp.item = beer bottle', while the action response after resolving should only be a 'succeeded' or 'aborted' message that is interpreted in SOAR as 'robot.object = beer bottle'.

In the current state of the system 10 different skills are implemented. The amount of productions checked in every loop step is of 77 rules.

It may happen that there is no plan for achieving the goal. In those situations SOAR implements several mechanisms to solve them:

- Subgoal capacity [21], allows the robot to find a way to get out of an impasse with the current actions available in order to achieve the desired state. This would be the case in which the robot could not decide the best action in the current situation with the available knowledge because there is no distinctive preference.
- Chunking ability [21][22][23], allows the production of new rules that help the robot adapt to new situations and, given a small set of primitive actions, execute full featured and specific goals never faced before.
- Reinforcement learning [24], together with the two previous features, helps the robot in learning to perform maintained goals such as keeping a room clean or learning by the use of user-defined heuristics in order to achieve, not only good results like using chunking, but near-optimal performances.

The two first mechanisms were activated for our approach. Use of the reinforcement learning will be analysed in future works. Those two mechanisms are specially important because thanks to them, the robot is capable of finding its own way to achieve any goal achievable with the current skills of the robot. Also, chunking makes decisions easier when the robot faces similar situations early experienced. This strengths allow the robot to adapt to new goals and situations without further programming than defining a goal or admit the expansion of its capabilities by simply defining a new skill.

2.4 Action Nodes

The action nodes are ROS software modules. They are modular pieces of software implemented to make the robot capable of performing each one of its abilities, defined in the SOAR module as the possible skill. Every time that SOAR proposes an skill to be performed calls the action node in charge of that skill.

When an action node is executed it provides some feedback to SOAR about its succees or failure. The feedback is captured by the interface and sent to SOAR in order to update the current state of the world.

The set of skills implemented and their associated actions are described in table 1

Skill	**Action**
Introduce himself	Talks about himself
Follow person	Follows a specific person in font of him
Search objects	Looks for objects in front of him
Search person	Looks for some person in the area
Grasp object	Grasps an specific object
Deliver object	Delivers an object to the person or place in front
Memorize person	Learns a person's face and stores his name
Exit apartment	Looks for the nearest exit and exits the area
Recognize person	Checks if the person in front was already known and retrieves its name
Point at an object	Points the location of an specific object

Table 1. Table of skills available at the robot and their associated actions

3 Platform: Reem

The robot platform used for testing the system developed is called Reem 2, a humanoid service robot created by PAL Robotics. Its weight is about 90 Kg, 22 degrees of freedom and an autonomy of about 8 hours. Reem is controlled by OROCOS for real time operations and by ROS for skill depletion. Among other abilities, it can recognize and grasp objects, detect faces, follow a person and even clean a room of objects that do not belong to it. In order to include robust grasping and gesture detection, a kinnect sensor on a headset on her head has been added to the commercial version.

Fig. 2. (a) Reem humanoid robot and (b) Reem head with kinect included

The robot is equipped with a Core 2 Duo and an ATOM computer, which provide all the computational power required to perform all tasks control. This means that all the algorithms required to plan and perform all the abilities are executed inside the robot.

4 Results

The whole architecture has been put to test in an environment that mimics that of the RoboCup@Home League at the GPSR test [25] (see figure 3). In this test, the robot has to listen three different types of commands with increased difficulty, and execute the required actions (skills) to accomplish the command. For our implementation, only the two first categories have been tested, as described in section 2.2.

Fig. 3. Reem robot at the experiments environment that mimics a home

Testing involved providing the robot with a spoken command, and checking that the robot was able to perform the required actions to complete the goal.

Examples of sentences the robot has been tested with (among others):

Category I *Go to the kitchen, find a coke and grasp it*
Sequence of actions performed by the robot:
understand command, go to kitchen, look for coke, grasp coke
Go to reception, find a person and introduce yourself
Sequence of actions performed by the robot:
understand command, go to reception, look for person, go to person, introduce yourself
Find the closest person, introduce yourself and follow the person in front of you
Sequence of actions performed by the robot:
look for a person, move to person, introduce yourself, follow person
Category II *Point at a seating*
Sequence of actions performed by the robot:

understand command, ask questions, acknowledge all information, navigate to location, search for seating, point at seating
Carry a Snack to a table
Sequence of actions performed by the robot:
understand command, ask questions, acknowledge all information, navigate to location, search for snack, grasp snack, go to table, deliver snack
Bring me an energy drink (figure 4)
Sequence of actions performed by the robot:
understand command, ask questions, acknowledge all information, navigate to location, search for energy drink, grasp energy drink, return to origin, deliver energy drink

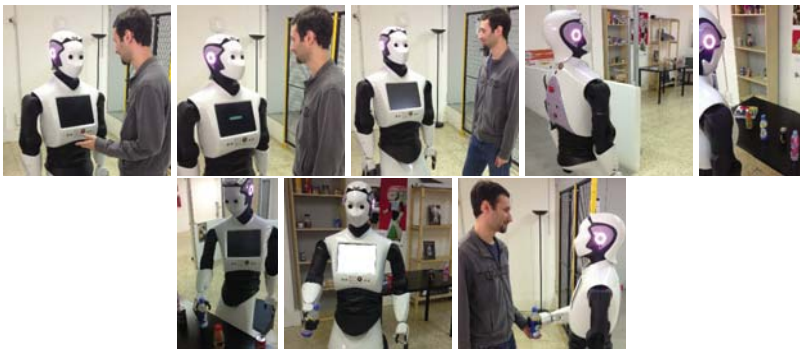

Fig. 4. Sequence of actions done by Reem to solve the command *Bring me an energy drink*

The system we present in this paper guarantees that the actions proposed will lead to the goal, so the robot will find a solution, although it can not be assured to be the optimal one. For instance, in some situations, the robot moved to a location that was not the correct one, before moving on a second action step to the correct one. However, the completion of the task is assured since the architecture will continue providing steps until the goal is accomplished.

5 Conclusions

The architecture presented allowed to command a commercial humanoid robot to perform a bunch of tasks as a combination of skills, without having to specify before hand how the skills have to be combined to solve the task. The whole approach avoids AI planning in the classical sense and uses instead a cognitive approach (SOAR) based on solving the current situation the robot faces. By solving the current situation skill by skill the robot finally achieves the goal (if

it is achievable). Given a goal and a set of skills, SOAR itself will generate the necessary steps to fulfil the goal using the skills (or at least try to reach the goal). Because of that, we can say that it can easily adapt to new goals effortlessly.

SOAR cannot detect if the goal requested to the robot is achiebable or not. If the goal is not achiebable, SOAR will keep trying to reach it, and send skill activations to the robot forever. In our implementation, the set of goals that one can ask the robot are restricted by the speech recognition system. Our system ensures that all the accepted vocal commands are achievable by a SOAR execution.

The whole architecture is completely robot agnostic, and can be adapted to any other robot provided that the skills are implemented and available to be called using the same interface. More than that, adding and removing skills becomes as simple as defining the conditions to work with them and their outcomes.

The current implementation can be improved in terms of robustness, solving two known issues:

First, if one of the actions is not completely achieved (for example, the robot is not able o reach a position in the space because it is occupied, or the robot cannot find an object that is in front of it), the skill activation will fail. However, in the current implementation the robot has no means to discover the reason of the failure. Hence the robot will detect that the state of the world has not changed, and hence select the same action (retry) towards the goal accomplishment. This behaviour could lead to an infinite loop of retries.

Second, this architecture is still not able to solve commands when errors in sentences are encountered (category III of the GPSR Robocup test). Future versions of the architecture will include this feature by including semantic and relation ontologies like Wordnet [26] and VerbNet [27], making this service robot more robust and general.

References

1. Haidegger, T., Barreto, M., Gonçalves, P., Habib, M.K., Ragavan, S.K.V., Li, H., Vaccarella, A., Perrone, R., Prestes, E.: Applied ontologies and standards for service robots. Robotics and Autonomous Systems (June 2013) 1–9
2. Stuart Russell, P.N.: Artificial Intelligence: A Modern Approach
3. Pollack, J.B.: Book Review : Allen Newell , Unified Theories of Cognition *
4. Jones, R.M.: An Introduction to Cognitive Architectures for Modeling and Simulation. (1987) (2004)
5. Kelley, T.D.: Developing a Psychologically Inspired Cognitive Architecture for Robotic Control : The Symbolic and Subsymbolic Robotic Intelligence Control System. Internation Journal of Advanced Robotic Systems **3**(3) (2006) 219–222
6. Langley, P., Laird, J.E., Rogers, S.: Cognitive architectures: Research issues and challenges. Cognitive Systems Research **10**(2) (June 2009) 141–160
7. Laird, J.E., Wray III, R.E.: Cognitive Architecture Requirements for Achieving AGI. In: Proceedings of the Third Conference on Artificial General Intelligence. (2010)

8. Chen, X., Ji, J., Jiang, J., Jin, G., Wang, F., Xie, J.: Developing High-level Cognitive Functions for Service Robots. AAMAS '10 Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems **1** (2010) 989–996
9. Laird, J.E., Kinkade, K.R., Mohan, S., Xu, J.Z.: Cognitive Robotics using the Soar Cognitive Architecture. In: Proc. of the 6th Int. Conf.on Cognitive Modelling. (2004) 226–230
10. Anderson, J.R.: ACT: A Simple Theory of Complex Cognition. American Psychologist (1995)
11. Stewart, T.C., West, R.L.: Deconstructing ACT-R. In: Proceedings of the Seventh International Conference on Cognitive Modeling. (2006)
12. Beetz, M., Lorenz, M., Tenorth, M.: CRAM A C ognitive R obot A bstract M achine for Everyday Manipulation in Human Environments. In: International Conference on Intelligent Robots and Systems (IROS). (2010)
13. Wei, C., Hindriks, K.V.: An Agent-Based Cognitive Robot Architecture. (2013) 54–71
14. Beetz, M., Klank, U., Kresse, I., Maldonado, A., Mösenlechner, L., Pangercic, D., Rühr, T., Tenorth, M.: Robotic Roommates Making Pancakes. In: 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia (October, 26–28 2011)
15. Hanford, S.D.: A Cognitive Robotic System Based on Soar. PhD thesis (2011)
16. Ravishankar, M.K.: Efficient algorithms for speech recognition. Technical report (1996)
17. Bird, S.: NLTK : The Natural Language Toolkit. In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. (2005) 1–4
18. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics **1** (2003) 423–430
19. Hall, J.: MaltParser An Architecture for Inductive Labeled Dependency Parsing. PhD thesis (2006)
20. Wintermute, S., Xu, J., Laird, J.E.: SORTS : A Human-Level Approach to Real-Time Strategy AI. (2007) 55–60
21. Laird, J.E., Newell, A., Rosenbloom, P.S.: SOAR: An Architecture for General Intelligence. Artificial Intelligence (1987)
22. Howes, A., Young, R.M.: The Role of Cognitive Architecture in Modelling the User : Soar s Learning Mechanism. (01222) (1996)
23. SoarTechnology: Soar : A Functional Approach to General Intelligence. Technical report (2002)
24. Nason, S., Laird, J.E.: Soar-RL : Integrating Reinforcement Learning with Soar. In: Cognitive Systems Research. (2004) 51–59
25. : Robocup@home rules and regulations
26. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11) (1995) 39–41
27. Palmer, M., Kipper, K., Korhonen, A., Ryant, N.: Extensive Classifications of English verbs. In: Proceedings of the 12th EURALEX International Congress. (2006)

Towards a Cognitive Architecture for Music Perception

Antonio Chella

Department of Chemical, Management, Computer, Mechanical Engineering
University of Palermo, Viale delle Scienze, building 6
90128 Palermo, Italy, antonio.chella@unipa.it

Abstract. The framework of a cognitive architecture for music perception is presented. The architecture extends and completes a similar architecture for computer vision developed during the years. The extended architecture takes into account many relationships between vision and music perception. The focus of the architecture resides in the intermediate area between the subsymbolic and the linguistic areas, based on conceptual spaces. A conceptual space for the perception of notes and chords is discussed along with its generalization for the perception of music phrases. A focus of attention mechanism scanning the conceptual space is also outlined. The focus of attention is driven by suitable linguistic and associative expectations on notes, chords and music phrases. Some problems and future works of the proposed approach are also outlined.

1 Introduction

Gärderfors [1], in his paper on "Semantics, Conceptual Spaces and Music" discusses a program for musical spaces analysis directly inspired to the framework of vision proposed by Marr [2]. More in details, the first level that feeds input to all the subsequent levels is related with *pitch identification*. The second level is related with the identification of *musical intervals*; this level takes also into account the cultural background of the listener. The third level is related with *tonality*, where scales are identified and the concepts of chromaticity and modulation arise. The fourth level of analysis is related with the interplay of pitch and time. According to Gärdenfors, time is concurrently processed by means of different levels related with *temporal intervals*, *beats*, *rhythmic patterns*, and at this level the analysis of pitch and the analysis of time merge together.

The correspondences between vision and music perception have been discussed in details by Tanguiane [3]. He considers three different levels of analysis distinguishing between statics and dynamics perception in vision and music. The first visual level in statics perception is the level of pixels, in analogy of the image level of Marr, that corresponds to the perception of *partials* in music. At the second level, the perception of simple patterns in vision corresponds to the perception of single *notes*. Finally at the third level, the perception of structured

patterns (as patterns of patterns), corresponds to the perception of *chords*. Concerning dynamic perception, the first level is the same as in the case of static perception, i.e., pixels vs. partials, while at the second level the perception of visual objects corresponds to the perception of musical notes, and at the third final level the perception of visual trajectories corresponds to the perception of music *melodies*.

Several cognitive models of music cognition have been proposed in the literature based on different symbolic or subsymbolic approaches, see Pearce and Wiggins [4] and Temperley [5] for recent reviews. Interesting systems, representative of these approaches are: MUSACT [6][7] based on various kinds of neural networks; the IDyOM project based on probabilistic models of perception [4][8][9]; the Melisma system [10] based on preference rules of symbolic nature; the HARP system, aimed at integrating symbolic and subsymbolic levels [11][12].

Here, we sketch a cognitive architecture for music perception that extends and completes an architecture for computer vision developed during the years. The proposed cognitive architecture integrates the symbolic and the sub symbolic approaches and it has been employed for static scenes analysis [13][14], dynamic scenes analysis [15], reasoning about robot actions [16], robot recognition of self [17] and robot self-consciousness [18]. The extended architecture takes into account many of the above outlined relationships between vision and music perception.

In analogy with Tanguiane, we distinguish between "static" perception related with the perception of chords in analogy with perception of static scenes, and "dynamic" perception related with the perception of musical phrases, in analogy with perception of dynamic scenes.

The considered cognitive architecture for music perception is organized in three computational areas - a term which is reminiscent of the cortical areas in the brain - that follows the Gärdenfors theory of *conceptual spaces* [19] (see Forth et al. [20] for a discussion on conceptual spaces and musical systems).

In the following, Section 2 outlines the cognitive architecture for music perception, while Section 3 describes the adopted music conceptual space for the perception of tones. Section 4 presents the linguistic area of the cognitive architecture and Section 5 presents the related operations of the focus of attention. Section 6 outlines the generalization of the conceptual space for tones perception to the case of perception of music phrases, and finally Section 7 discusses some problems of the proposed approach and future works.

2 The Cognitive Architecture

The proposed cognitive architecture for music perception is sketched in Figure 1. The areas of the architecture are concurrent computational components working together on different commitments. There is no privileged direction in the flow of information among them: some computations are strictly bottom-up, with data flowing from the subconceptual up to the linguistic through the conceptual area; other computations combine top-down with bottom-up processing.

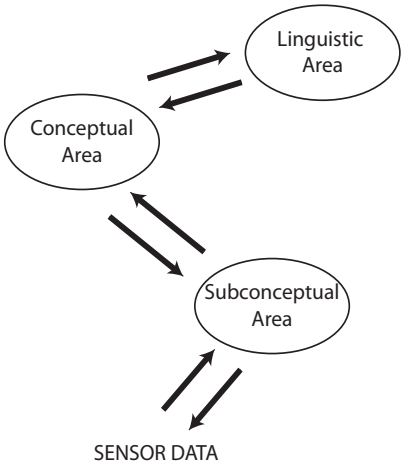

Fig. 1. A sketch of the cognitive architecture.

The *subconceptual* area of the proposed architecture is concerned with the processing of data directly coming from the sensors. Here, information is not yet organized in terms of conceptual structures and categories. In the *linguistic* area, representation and processing are based on a logic-oriented formalism.

The *conceptual* area is an intermediate level of representation between the subconceptual and the linguistic areas and based on conceptual spaces. Here, data is organized in conceptual structures, that are still independent of linguistic description. The symbolic formalism of the linguistic area is then interpreted on aggregation of these structures.

It is to be remarked that the proposed architecture cannot be considered as a model of human perception. No hypotheses concerning its cognitive adequacy from a psychological point of view have been made. However, various cognitive results have been taken as sources of inspiration.

3 Music Conceptual Space

The conceptual area, as previously stated, is the area between the subconceptual and the linguistic area, and it is based on conceptual spaces. We adopt the term *knoxel* (in analogy with the term *pixel*) to denote a point in a conceptual space CS. The choice of this term stresses the fact that a point in CS is the knowledge primitive element at the considered level of analysis.

The conceptual space acts as a workspace in which low-level and high-level processes access and exchange information respectively from bottom to top and from top to bottom. However, the conceptual space has a precise geometric structure of metric space and also the operations in CS are geometric ones: this

structure allows us to describe the functionalities of the cognitive architecture in terms of the language of geometry.

In particular, inspired by many empirical investigations on the perception of tones (see Oxenham [21] for a review) we adopt as a knoxel of a *music conceptual space* the set of partials of a perceived tone. A knoxel $\mathbf{k}$ of the music CS is therefore a vector of the main perceived partials of a tone in terms of the Fourier Transform analysis. A similar choice has been carried out by Tanguiane [3] concerning his proposed *correlativity* model of perception.

It should be noticed that the partials of a tone are related both with the pitch and the timbre of the perceived note. Roughly, the fundamental frequency is related with the pitch, while the amplitude of the remaining partials are also related with the timbre of the note. By an analogy with the case of static scenes analysis, a knoxel changes its position in CS when a perceived 3D primitive changes its position in space or its shape [13]; in the case of music perception, the knoxel in the music CS changes its position either when the perceived sound changes its pitch or its timbre changes as well. Moreover, considering the partials of a tone allows us to deal also with microtonal tones, trills, embellished notes, rough notes, and so on.

A *chord* is a set of two or more tones perceived at the same time. The chord is treated as a complex object, in analogy with static scenes analysis where a complex object is an object made up by two or more 3D primitives. A chord is then represented in music CS as the set of the knoxels $[\mathbf{k}_a, \mathbf{k}_b, \dots]$ related with the constituent tones. It should be noticed that the tones of a chord may differ not only in pitch, but also in timbre. Figure 2 is an evocative representation of a chord in the music CS made up by knoxel $\mathbf{k}_a$ corresponding to tone C and knoxel $\mathbf{k}_b$ corresponding to the tone G.

In the case of perception of complex objects in vision, their mutual positions and shapes are important in order to describe the perceived object: e.g., in the case of an hammer, the mutual positions and the mutual shapes of the handle and the head are obviously important to classify the composite object as an hammer. In the same way, the mutual relationships between the pitches (and the timbres) of the perceived tones are important in order to describe the perceived chord. Therefore, spatial relationships in static scenes analysis are in some sense analogous to sounds relationships in music CS.

It is to be noticed that this approach allows us to represent a chord as a set of knoxels in music CS. In this way, the cardinality of the conceptual space does not change with the number of tones forming the chord. In facts, all the tones of the chord are perceived at the same time but they are represented as different points in the same music CS; that is, the music CS is a sort of *snapshot* of the set of the perceived tones of the chord.

In the case of a temporal progression of chords, a scattering occur in the music CS: some knoxels which are related with the same tones between chords will remain in the same position, while other knoxels will change their position in CS, see Figure 3 for an evocative representation of scattering in the music CS. In the figure, the knoxels $\mathbf{k}_a$, corresponding to C, and $\mathbf{k}_b$, corresponding to E,

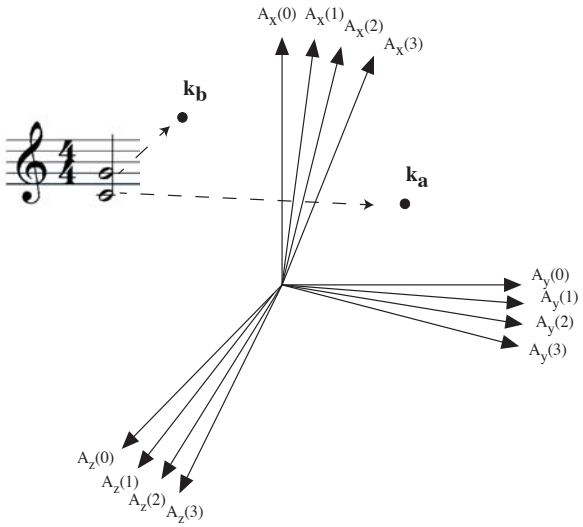

Fig. 2. An evocative representation of a chord in the *music conceptual space*.

change their position in the new chord: they becomes A and D, while knoxel $\mathbf{k}_c$, corresponding to G, maintains its position. The relationships between mutual positions in music CS could then be employed to analyze the chords progression and the relationships between subsequent chords.

A problem may arise at this point. In facts, in order to analyze the progression of chords, the system should be able to find the correct correspondences between subsequent knoxels: i.e., $\mathbf{k}'_a$ should correspond to $\mathbf{k}_a$ and not to, e.g., $\mathbf{k}_b$. This is a problem similar to the *correspondence* problem in stereo and in visual motion analysis: a vision system analyzing subsequent frames of a moving object should be able to find the correct corresponding object tokens among the motion frames; see the seminal book by Ullman [22] or Chap. 11 of the recent book by Szeliski [23] for a review. However, it should be noticed that the expectation generation mechanism described in Section 5 could greatly help facing this difficult problem.

The described representation is well suited for the recognition of chords: for example we may adopt the algorithms proposed by Tanguiane [3]. However, Tanguiane hypothesizes, at the basis of his *correlativity* principle, that all the notes of a chord have the same shifted partials, while we consider the possibility that a chord could be made by tones with different partials.

The proposed representation is also suitable for the analysis of the efficiency in *voice leading*, as described by Tymoczko [24]. Tymoczko describes a geometrical analysis of chords by considering several spaces with different cardinalities,

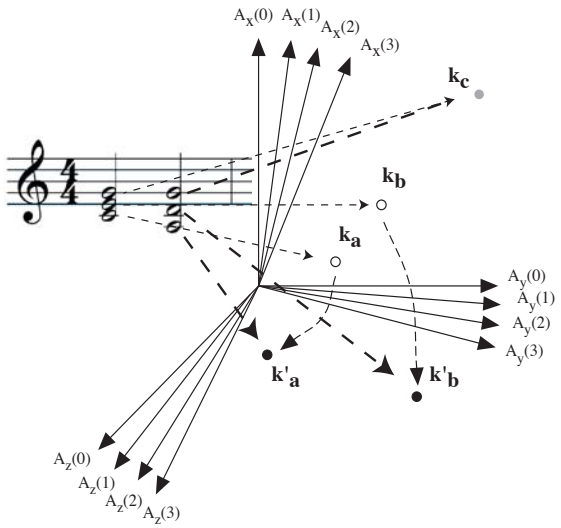

Fig. 3. An evocative representation of a scattering between two chords in the *music conceptual space*.

i.e., a one note circular space, a two note space, a three note space, and so on. Instead, the cardinality of the considered conceptual space does not change, as previously remarked.

4 Linguistic area

In the linguistic area, the representation of perceived tones is based on a high level, logic oriented formalism. The linguistic area acts as a sort of long term memory, in the sense that it is a semantic network of symbols and their relationships related with musical perceptions. The linguistic area also performs inferences of symbolic nature. In preliminary experiments, we adopted a linguistic area based on a hybrid KB in the KL-ONE tradition [25]. A hybrid formalism in this sense is constituted by two different components: a *terminological* component for the description of concepts, and an *assertional* component, that stores information concerning a specific context. A similar formalism has been adopted by Camurri et al. in the HARP system [11][12].

In the domain of perception of tones, the terminological component contains the description of relevant concepts such as chords, tonic, dominant and so on. The assertional component stores the assertions describing specific situations. Figure 4 shows a fragment of the terminological knowledge base along with its mapping into the corresponding entities in the conceptual space.

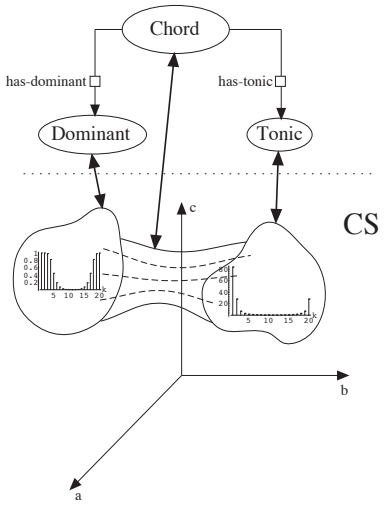

Fig. 4. A fragment of the terminological KB along with its mapping into the conceptual space.

A generic *Chord* is described as composed of at least two knoxels. A *Simple-Chord* is a chord composed by two knoxels; a *Complex-Chord* is a chord composed of more than two knoxels. In the considered case, the concept *Chord* has two roles: a role *has-dominant*, and a role *has-tonic* both filled with specific tones.

In general, we assume that the description of the concepts in the symbolic KB is not exhaustive. We symbolically represent the information necessary to make suitable inferences.

The assertional component contains facts expressed as assertions in a predicative language, in which the concepts of the terminological components correspond to one argument predicates, and the roles (e.g., part_of) correspond to two argument relations. For example, the following predicates describe that the instance `f7#1` of the F7 chord has a dominant which is the constant `ka` corresponding to a knoxel $\mathbf{k}_a$ and a tonic which is the constant `k#b` corresponding to a knoxel $\mathbf{k}_b$ of the current CS:

```
ChordF7(f7#1)
has-dominant(f7#1,ka)
has-tonic(f7#1,kb)
```

By means of the mapping between symbolic KB and conceptual spaces, the linguistic area assigns names (symbols) to perceived entities, describing their structure with a logical-structural language. As a result, all the symbols in the linguistic area find their meaning in the conceptual space which is inside the system itself.

A deeper account of these aspects can be found in Chella et at. [13].

5 Focus of Attention

A cognitive architecture with bounded resources cannot carry out a one-shot, exhaustive, and uniform analysis of the perceived data within reasonable resource constraints. Some of the perceived data (and of the relations among them) are more relevant than others, and it should be a waste of time and of computational resources to detect true but useless details.

In order to avoid the waste of computational resources, the association between symbolic representations and configurations of knoxels in CS is driven by a sequential scanning mechanism that acts as some sort of internal focus of attention, and inspired by the attentive processes in human perception.

In the considered cognitive architecture for music perception, the perception model is based on a focus of attention that selects the relevant aspects of a sound by sequentially scanning the corresponding knoxels in the conceptual space. It is crucial in determining which assertions must be added to the linguistic knowledge base: not all true (and possibly useless) assertions are generated, but only those that are judged to be relevant on the basis of the attentive process.

The recognition of a certain component of a perceived configuration of knoxels in music CS will elicit the *expectation* of other possible components of the same chord in the perceived conceptual space configuration. In this case, the mechanism seeks for the corresponding knoxels in the current CS configuration. We call this type of expectation *synchronic* because it refers to a single configuration in CS.

The recognition of a certain configuration in CS could also elicit the expectation of a scattering in the arrangement of the knoxels in CS; i.e., the mechanism generates the expectations for another set of knoxels in a subsequent CS configuration. We call this expectation *diachronic*, in the sense that it involves subsequent configurations of CS. Diachronic expectations can be related with progression of chords. For example, in the case of jazz music, when the system recognized the $Cmajor$ key (see Rowe [26] for a catalogue of key induction algorithms) and a Dm chord is perceived, then the focus of attention will generate the expectations of G and C chords in order to search for the well known chord progression $ii - V - I$ (see Chap. 10 of Tymoczko [24]).

Actually, we take into account two main sources of expectations. On the one side, expectations could be generated on the basis of the structural information stored in the symbolic knowledge base, as in the previous example of the jazz chord sequence. We call these expectations *linguistic*. Several sources may be taken into account in order to generate linguistic expectations, for example the *ITPRA* theory of expectation proposed by Huron [27], the preference rules systems discussed by Temperley [10] or the rules of harmony and voice leading discussed in Tymoczko [24], just to cite a few. As an example, as soon as a particular configuration of knoxel is recognized as a possible chord filling the role

of the first chord of the progression $ii - V - I$, the symbolic KB generates the expectation of the remaining chords of the sequence.

On the other side, expectations could be generated by purely Hebbian, associative mechanisms. Suppose that the system learnt that typically a jazz player adopts the *tritone* substitution when performing the previous described jazz progression. The system could learn to associate this substitution to the progression: in this case, when a compatible chord is recognized, the system will generate also expectations for the sequence $ii - \flat II - I$. We call these expectations *associative*.

Therefore, synchronic expectations refer to the same configuration of knoxels at the same time; diachronic expectations involve subsequent configurations of knoxels. The linguistic and associative mechanisms let the cognitive architecture generate suitable expectations related to the perceived chords progressions.

6 Perception of Music Phrases

So far we adopted a "static" conceptual space where a knoxel represents the partials of a perceived tone. In order to generalize this concept and in analogy with the differences between static and dynamic vision, in order to represent a music *phrase*, we now adopt a "dynamic" conceptual space in which each knoxel represents the whole set of partials of the Short Time Fourier Transform of the corresponding music phrase. In other words, a knoxel in the dynamic CS now represents all the parameters of the *spectrogram* of the perceived phrase.

Therefore, inspired by empirical results (see Deutsch [28] for a review) we hypothesize that a musical phrase is perceived as a whole "Gestaltic" group, in the same way as a movement could be visually perceived as a whole and not as a sequence of single frames. It should be noticed that, similarly to the static case, a knoxel represents the sequence of pitches and durations of the perceived phrase and also its timbre: the same phrase played by two different instruments corresponds to two different knoxels in the dynamic CS.

The operations in the dynamic CS are largely similar to the static CS, with the main difference that now a knoxel is a whole perceived phrase.

A configuration of knoxels in CS occurs when two or more phrases are perceived at the same time. The two phrases may be related with two different sequences of pitches or it may be the same sequence played for example, by two different instruments. This is similar to the situation depicted in Figure 2, where the knoxels $\mathbf{k}_a$ and $\mathbf{k}_b$ are interpreted as music phrases perceived at the same time.

A scattering of knoxels occurs when a change occurs in a perceived phrase. We may represent this scattering in a similar way to the situation depicted in Figure 3, where the knoxels also in this case are interpreted as music phrases: knoxels $\mathbf{k}_a$ and $\mathbf{k}_b$ are interpreted as changed music phrases while knoxels $\mathbf{k}_c$ corresponds to the same perceived phrase.

As an example, let us consider the well known piece *In C* by Terry Riley. The piece is composed by 53 small phrases to be performed sequentially; each player may decide when to start playing, how many times to repeat the same

phrase, and when to move to the next phrase (see the performing directions of *In C* [29]).

Let us consider the case in which two players, with two different instruments, start with the first phrase. In this case, two knoxels $\mathbf{k}_a$ and $\mathbf{k}_b$ will be activated in the dynamic CS. We remark that, although the phrase is the same in terms of pitch and duration, it corresponds to two different knoxels because of different timbres of the two instruments. When a player will decide at some time to move to next phrase, a scattering occur in the dynamic CS, analogously with the previous analyzed static CS: the corresponding knoxel, say $\mathbf{k}_a$, will change its position to $\mathbf{k}'_a$.

The focus of attention mechanism will operate in a similar way as in the static case: the synchronous modality of the focus of attention will take care of generation of expectations among phrases occurring at the same time, by taking into account, e.g., the rules of counterpoint. Instead, the asynchronous modality will generate expectations concerning, e.g., the continuation of phrases.

Moreover, the static CS and the dynamic CS could generate mutual expectations: for example, when the focus of attention recognizes a progression of chords in the static CS, this recognized progression will constraint the expectations of phrases in the dynamic CS. As another example, the recognition of a phrase in the dynamic CS could constraint as well the recognition of the corresponding progression of chords in the static CS.

7 Discussion and Conclusions

The paper sketched a cognitive architecture for music perception extending and completing a computer vision cognitive architecture. The architecture integrates symbolic and the sub symbolic approaches by means of *conceptual spaces* and it takes into account many relationships between vision and music perception.

Several problems arise concerning the proposed approach. A first problem, analogously with the case of computer vision, concerns the *segmentation* step. In the case of static CS, the cognitive architecture should be able to segment the Fourier Transform signal coming from the microphone in order to individuate the perceived tones; in the case of dynamic CS the architecture should be able to individuate the perceived phrases. Although many algorithms for music segmentation have been proposed in the computer music literature and some of them are also available as commercial program, as the AudioSculpt program developed by IRCAM[1], this is a main problem in perception. Interestingly, empirical studies concur in indicating that the same Gestalt principles at the basis of visual perception operate in similar ways in music perception, as discussed by Deutsch [28].

The expectation generation process at the basis of the focus of attention mechanism can be employed to help solving the segmentation problem: the linguistic information and the associative mechanism can provide interpretation

[1] http://forumnet.ircam.fr/product/audiosculpt/

contexts and high level hypotheses that help segmenting the audio signal, as e.g., in the IPUS system [30].

Another problem is related with the analysis of *time*. Currently, the proposed architecture does not take into account the metrical structure of the perceived music. Successive development of the described architecture will concern a metrical conceptual space; interesting starting points are the geometric models of metrical-rhythmic structure discussed by Forth et al. [20].

However, we maintain that an intermediate level based on conceptual spaces could be a great help towards the integration between the music cognitive systems based on subsymbolic representations, and the class of systems based on symbolic models of knowledge representation and reasoning. In facts, conceptual spaces could offer a theoretically well founded approach to the integration of symbolic musical knowledge with musical neural networks.

Finally, as stated during the paper, the synergies between music and vision are multiple and multifaceted. Future works will deal with the exploitation of conceptual spaces as a framework towards a sort of *unified* theory of perception able to integrate in a principled way vision and music perception.

References

1. Gärdenfors, P.: Semantics, conceptual spaces andthe dimensions of music. In Rantala, V., Rowell, L., Tarasti, E., eds.: Essays on the Philosophy of Music. Philosophical Society of Finland, Helsinki (1988) 9–27
2. Marr, D.: Vision. W.H. Freeman and Co., New York (1982)
3. Tanguiane, A.: Artificial Perception and Music Recognition. Number 746 in Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin Heidelberg (1993)
4. Wiggins, G., Pearce, M., Müllensiefen: Computational modelling of music cognition and musical creativity. In Dean, R., ed.: The Oxford Handbook of Computer Music. Oxford University Press, Oxford (2009) 387–414
5. Temperley, D.: Computational models of music cognition. In Deutsch, D., ed.: The Psychology of Music. Third edn. Academic Press, Amsterdam, The Netherlands (2012) 327–368
6. Bharucha, J.: Music cognition and perceptual facilitation: A connectionist framework. Music Perception: An Interdisciplinary Journal **5**(1) (1987) 1–30
7. Bharucha, J.: Pitch, harmony and neural nets: A psychological perspective. In Todd, P., Loy, D., eds.: Music and Connectionism. MIT Press, Cambridge, MA (1991) 84–99
8. Pearce, M., Wiggins, G.: Improved methods for statistical modelling of monophonic music. Journal of New Music Research **33**(4) (2004) 367–385
9. Pearce, M., Wiggins, G.: Expectation in melody: The influence of context and learning. Music Perception: An Interdisciplinary Journal **23**(5) (2006) 377–406
10. Temperley, D.: The Cognition of Basic Musical Structures. MIT Press, Cambridge, MA (2001)
11. Camurri, A., Frixione, M., Innocenti, C.: A cognitive model and a knowledge representation system for music and multimedia. Journal of New Music Research **23** (1994) 317–347

12. Camurri, A., Catorcini, A., Innocenti, C., Massari, A.: Music and multimedia knowledge representation and reasoning: the HARP system. Computer Music Journal **19**(2) (1995) 34–58
13. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for artificial vision. Artificial Intelligence **89** (1997) 73–111
14. Chella, A., Frixione, M., Gaglio, S.: An architecture for autonomous agents exploiting conceptual representations. Robotics and Autonomous Systems **25**(3-4) (1998) 231–240
15. Chella, A., Frixione, M., Gaglio, S.: Understanding dynamic scenes. Artificial Intelligence **123** (2000) 89–132
16. Chella, A., Gaglio, S., Pirrone, R.: Conceptual representations of actions for autonomous robots. Robotics and Autonomous Systems **34** (2001) 251–263
17. Chella, A., Frixione, M., Gaglio, S.: Anchoring symbols to conceptual spaces: the case of dynamic scenarios. Robotics and Autonomous Systems **43**(2-3) (2003) 175–188
18. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for robot self-consciousness. Artificial Intelligence in Medicine **44** (2008) 147–154
19. Gärdenfors, P.: Conceptual Spaces. MIT Press, Bradford Books, Cambridge, MA (2000)
20. Forth, J., Wiggins, G., McLean, A.: Unifying conceptual spaces: Concept formation in musical creative systems. Minds and Machines **20** (2010) 503–532
21. Oxenham, A.: The perception of musical tones. In Deutsch, D., ed.: The Psychology of Music. Third edn. Academic Press, Amsterdam, The Netherlands (2013) 1–33
22. Ullman, S.: The Interpretation of Visual Motion. MIT Press, Cambridge,MA (1979)
23. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer, London (2011)
24. Tymoczko, D.: A Geometry of Music. Harmony and Counterpoint in the Extended Common Practice. Oxford University Press, Oxford (2011)
25. Brachman, R., Schmoltze, J.: An overview of the KL-ONE knowledge representation system. Cognitive Science **9**(2) (1985) 171–216
26. Rowe, R.: Machine Musicianship. MIT Press, Cambridge, MA (2001)
27. Huron, D.: Sweet Anticipation. Music and the Psychology of Expectation. MIT Press, Cambridge, MA (2006)
28. Deutsch, D.: Grouping mechanisms in music. In Deutsch, D., ed.: The Psychology of Music. Third edn. Academic Press, Amsterdam, The Netherlands (2013) 183–248
29. Riley, T.: *In C*: Performing directions. Celestial Harmonies (1964)
30. Lesser, V., Nawab, H., Klassner, F.: IPUS: An architecture for the integrated processing and understanding of signals. Artificial Intelligence **77** (1995) 129–171

Typicality-Based Inference by Plugging Conceptual Spaces Into Ontologies

Leo Ghignone, Antonio Lieto, and Daniele P. Radicioni

Università di Torino, Dipartimento di Informatica, Italy
{lieto,radicion}@di.unito.it
leo.ghignone@gmail.com

Abstract. In this paper we present a cognitively inspired system for the representation of conceptual information in an ontology-based environment. It builds on the heterogeneous notion of concepts in Cognitive Science and on the so-called *dual process theories* of reasoning and rationality, and it provides a twofold view on the same artificial concept, combining a classical symbolic component (grounded on a formal ontology) with a typicality-based one (grounded on the conceptual spaces framework). The implemented system has been tested in a pilot experimentation regarding the classification task of linguistic stimuli. The results show that this modeling solution extends the representational and reasoning "conceptual" capabilities of standard ontology-based systems.

1 Introduction

Representing and reasoning on *common sense* concepts is still an open issue in the field of knowledge engineering and, more specifically, in that of formal ontologies. In Cognitive Science evidences exist in favor of prototypical concepts, and typicality-based conceptual reasoning has been widely studied. Conversely, in the field of computational models of cognition, most contemporary concept oriented knowledge representation (KR) systems, including formal ontologies, do not allow –for technical convenience– neither the representation of concepts in prototypical terms nor forms of approximate, non monotonic, conceptual reasoning. In this paper we focus on the problem of concept representation in the field of formal ontologies and we introduce, following the approach proposed in [1], a cognitively inspired system to extend the representational and reasoning capabilities of the ontology based systems.

The study of concept representation concerns different research areas, such as Artificial Intelligence, Cognitive Science, Philosophy, etc.. In the field of Cognitive Science, the early work of Rosch [2] showed that ordinary concepts do not obey the classical theory (stating that concepts can be defined in terms of sets of necessary and sufficient conditions). Rather, they exhibit *prototypical* traits: e.g., some members of a category are considered *better instances* than other ones; more *central* instances share certain typical features –such as the ability of flying for birds– that, in general, cannot be thought of as necessary nor sufficient conditions. These results influenced pioneering KR research, where some efforts

were invested in trying to take into account the suggestions coming from Cognitive Psychology: artificial systems were designed –e.g., frames [3]– to represent and to conduct reasoning on concepts in "non classical", prototypical terms [4].

However, these systems lacked in clear formal semantics, and were later sacrificed in favor of a class of formalisms stemmed from structured inheritance semantic networks: the first system in this line of research was the KL-ONE system [5]. These formalisms are known today as description logics (DLs). In this setting, the representation of prototypical information (and therefore the possibility of performing non monotonic reasoning) is not allowed,[1] since the formalisms in this class are primarily intended for deductive, logical inference. Nowadays, DLs are largely adopted in diverse application areas, in particular within the area of ontology representation. For example, OWL and OWL 2 formalisms follow this tradition,[2] which has been endorsed by the W3C for the development of the Semantic Web. However, under a historical perspective, the choice of preferring classical systems based on a well defined –Tarskian-like– semantics left unsolved the problem of representing concepts in prototypical terms. Although in the field of logic oriented KR various fuzzy and non-monotonic extensions of DL formalisms have been designed to deal with some aspects of "non-classical" concepts, nonetheless various theoretical and practical problems remain unsolved [6].

As a possible way out, we follow the proposal presented in [1], that relies on two main cornerstones: the dual process theory of reasoning and rationality [7,8,9], and the heterogeneous approach to the concepts in Cognitive Science [10]. This paper has the following major elements of interest: *i)* we provided the hybrid architecture envisioned in [1] with a working implementation; *ii)* we show how the resulting system is able to perform a simple form of categorization, that would be unfeasible by using only formal ontologies; *iii)* we a propose a novel access strategy (different from that outlined in [1]) to the conceptual information, closer to the tenets of the dual process approach (more about this point later on).

The paper is structured as follows: in Section 2 we illustrate the general architecture and the main features of the implemented system. In Section 3 we provide the results of a preliminary experimentation to test inference in the proposed approach, and, finally, we conclude by presenting the related work (Section 4) and by outlining future work (Section 5).

2 The System

A system has been implemented to explore the hypothesis of the hybrid conceptual architecture. To test it, we have been considering a basic *inference* task: given an input description in natural language, the system should be able to find,

[1] This is the case, for example, of *exceptions* to the inheritance mechanism.

[2] For the Web Ontology Language, see `http://www.w3.org/TR/owl-features/` and `http://www.w3.org/TR/owl2-overview/`, respectively.

even for typicality based description (that is, most of common sense descriptions), the corresponding concept category by combining ontological inference and typicality based one. To these ends, we developed a domain ontology (the *naive animal ontology*, illustrated below) and a parallel typicality description as a set of domains in a conceptual space framework [11].

In the following, *i)* we first outline the design principles that drove the development of the system; *ii)* we then provide an overview of the system architecture and of its components and features; *iii)* we elaborate on the inference task, providing the detailed control strategy; and finally *iv)* we introduce the domain ontology and the conceptual space used as case study applied over the restricted domain of animals.

2.1 Background and architecture design

The theoretical framework known as *dual process theory* postulates the coexistence of two different types of cognitive systems. The systems[3] of the first type (type 1) are phylogenetically older, unconscious, automatic, associative, parallel and fast. The systems of the second type (type 2) are more recent, conscious, sequential and slow, and featured by explicit rule following [7,8,9]. According to the reasons presented in [12,1], the conceptual representation of our systems should be equipped with two major sorts of components, based on:

- type 1 processes, to perform fast and approximate categorization by taking advantage from prototypical information associated to concepts;
- type 2 processes, involved in complex inference tasks and that do not take into account the representation of prototypical knowledge.

Another theoretical framework inspiring our system regards the heterogeneous approach to the concepts in Cognitive Science, according to which concepts do not constitute a unitary element (see [10]).

Our system is equipped, then, with a hybrid conceptual architecture based on a classical component and on a typical component, each encoding a specific reasoning mechanism as in the dual process perspective. Figure 1 shows the general architecture of the hybrid conceptual representation.

The ontological component is based on a classical representation grounded on a DL formalism, and it allows specifying the necessary and/or sufficient conditions for concept definition. For example, if we consider the concept *water*, the classical component will contain the information that *water* is exactly the chemical substance whose formula is H_2O, i.e., the substance whose molecules have two hydrogen atoms with a covalent bond to the single oxygen atom. On the other hand, the prototypical facet of the concept will grasp its prototypical traits, such as the fact that water occurring in liquid state is usually a colorless, odorless and tasteless fluid.

[3] We assume that each system type can be composed by many sub-systems and processes.

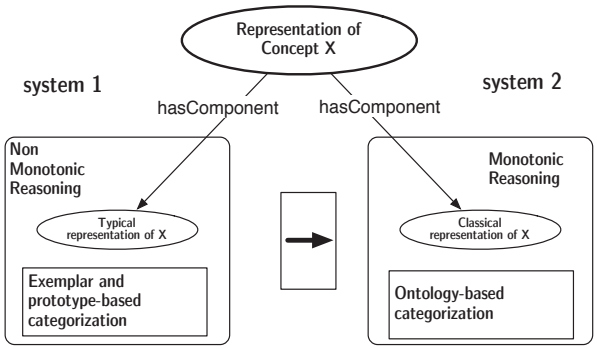

Fig. 1: Architecture of the hybrid system.

By adopting the "dual process" notation, in our system the representational and reasoning functions are assigned to the system 1 (executing processes of type 1), and they are associated to the Conceptual Spaces framework [11]. Both from a modeling and from a reasoning point of view, system 1 is compliant with the traits of conceptual typicality. On the other hand, the representational and reasoning functions assigned to the system 2 (executing processes of type 2) are associated to a classical DL-based ontological representation. Differently from what proposed in [1], the access to the information stored and processed in both components is assumed to proceed from the system 1 to the system 2, as suggested by the central arrow in Figure 1.

We now briefly introduce the representational frameworks upon which system 1 (henceforth $\mathcal{S}1$) and system 2 (henceforth $\mathcal{S}2$) have been designed.

As mentioned, the aspects related to the typical conceptual component $\mathcal{S}1$ are modeled through Conceptual Spaces [11]. Conceptual spaces (CS) are a geometrical framework for the representation of knowledge, consisting in a set of *quality dimensions*. In some cases, such dimensions can be directly related to perceptual mechanisms; examples of this kind are temperature, weight, brightness, pitch. In other cases, dimensions can be more abstract in nature. A geometrical (topological or metrical) structure is associated to each quality dimension. The chief idea is that knowledge representation can benefit from the geometrical structure of conceptual spaces: instances are represented as points in a space, and their similarity can be calculated in the terms of their distance according to some suitable distance measure. In this setting, concepts correspond to regions, and regions with different geometrical properties correspond to different kinds of concepts. Conceptual spaces are suitable to represent concepts in "typical" terms, since the regions representing concepts have soft boundaries. In many cases typicality effects can be represented in a straightforward way: for example, in the case of concepts, corresponding to convex regions of a conceptual space, prototypes have a natural geometrical interpretation, in that they correspond to the geometrical centre of the region itself. Given a convex region, we can

provide each point with a certain centrality degree, that can be interpreted as a measure of its typicality. Moreover, single exemplars correspond to single points in the space. This allows us to consider both the exemplar and the prototypical accounts of typicality (further details can be found in [13, p. 9]).

On the other hand, the representation of the classical component $\mathcal{S}2$ has been implemented based on a formal ontology. As already pointed out, the standard ontological formalisms leave unsolved the problem of representing prototypical information. Furthermore, it is not possible to execute non monotonic inference, since classical ontology-based reasoning mechanisms simply contemplate deductive processes.

2.2 Inference in the hybrid system

Categorization (i.e., to classify a given data instance into a predefined set of categories) is one of the classical processes automatically performed both by symbolic and sub-symbolic artificial systems. In our system categorization is based on a two-step process involving both the typical and the classical component of the conceptual representation. These components account for different types of categorization: approximate or non monotonic (performed on the conceptual spaces), and classical or monotonic (performed on the ontology). Different from classical ontological inference, in fact, categorization in conceptual spaces proceeds from *prototypical* values. In turn, prototypical values need not be specified for all class individuals, that vice versa can overwrite them: one typical example is the case of birds that (by default) fly, except for special birds, like penguins, that do not fly.

The whole categorization process regarding our system can be summarized as follows. The system takes in input a textual description d and produces in output a pair of categories $\langle \mathsf{c_0}, \mathsf{cc} \rangle$, the output of $\mathcal{S}1$ and $\mathcal{S}2$, respectively. The $\mathcal{S}1$ component takes in input the information extracted from the description d, and produces in output a set of classes $\mathsf{C} = \{\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_n\}$. This set of results is then checked against cc, the output of $\mathcal{S}2$ (Algorithm 1, line 3): the step is performed by adding to the ontology an individual from the class $\mathsf{c_i} \in \mathsf{C}$, modified by the information extracted from d, and by checking the consistency of the newly added element with a DL reasoner.

If the $\mathcal{S}2$ system classifies it as consistent with the ontology, then the classification succeeded and the category provided by $\mathcal{S}2$ (cc) is returned along with $\mathsf{c_0}$, the top scoring class returned by $\mathcal{S}1$ (Algorithm 1: line 8). If cc –the class computed by $\mathcal{S}2$– is a superclass or a subclass of one of those identified by $\mathcal{S}1$ ($\mathsf{c_i}$), both cc and $\mathsf{c_0}$ are returned (Algorithm 1: line 11). Thus, if $\mathcal{S}2$ provides more specific output, we follow a *specificity* heuristics; otherwise, the output of $\mathcal{S}2$ is returned, following the rationale that it is *safer*.[4] If all results in C are

[4] The output of $\mathcal{S}2$ cannot be wrong on a purely logical perspective, in that it is the result of a deductive process. The control strategy tries to implement a tradeoff between ontological inference and the output of $\mathcal{S}1$, which is more informative but also less reliable from a formal point of view. However, in next future we plan to explore different conciliation mechanisms to ground the overall control strategy.

Algorithm 1 Inference in the hybrid system.

input : textual description d
output : a class assignment, as computed by $\mathcal{S}1$ and $\mathcal{S}2$

```
1: C ← S1(d) /* conceptual spaces output */
2: for each c_i ∈ C do
3:    cc ← S2(⟨d, c_i⟩) /* ontology based output */
4:    if cc == NULL then
5:       continue /* inconsistency detected */
6:    end if
7:    if cc equals c_i then
8:       return ⟨c_0, cc⟩
9:    else
10:      if cc is subclass or superclass of c_i then
11:         return ⟨c_0, cc⟩
12:      end if
13:   end if
14: end for
15: cc ← S2(⟨d, Thing⟩)
16: return ⟨c_0, cc⟩
```

inconsistent with those computed by $\mathcal{S}2$, a pair of classes is returned including c_0 and the output of $\mathcal{S}2$ having for actual parameters d and Thing, the meta class of all the classes in the ontological formalism.

2.3 Developing the Ontology

A formal ontology has been developed describing the animal kingdom. It has been devised to meet common sense intuitions, rather than reflecting the precise taxonomic knowledge of ethologists, so we denote it as *naïve animal ontology*.[5] In particular, the ontology contains the taxonomic distinctions that have an intuitive counterpart in the way human beings categorize the corresponding concepts. Classes are collapsed at a granularity level such that they can be naturally grouped together also based on their *accessibility* [14]. For example, although the category *pachyderm* is no longer in use by ethologists, we created a *pachyderm* class that is superclass to *elephant*, *hippopotamus*, and *rhinoceros*. The underlying *rationale* is that it is still in use by non experts, due to the intuitive resemblances among its subclasses.

The ontology is linked to DOLCE's *Lite* version;[6] in particular, the tree containing our taxonomy is rooted in the *agentive-physical-object* class, while the body components are set under *biological-physical-object*, and partitioned between the two disjunct classes *head-part* (e.g., for framing horns, antennas, fang, etc.) and *body-part* (e.g., for paws, tails, etc.). The *biological-object* class in-

[5] The ontology is available at the URL `http://www.di.unito.it/~radicion/datasets/aic_13/Naive_animal_ontology.owl`
[6] `http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl`

cludes different sorts of skins (such as *fur*, *plumage*, *scales*), substances produced and eaten by animals (e.g., *milk*, *wool*, *poison* and *fruits*, *leaves* and *seeds*).

2.4 Formalizing conceptual spaces and distance metrics

The conceptual space defines a metric space that can be used to compute the proximity of the input entities to prototypes. To compute the distance between two points p_1, p_2 we apply a distance metrics based on the combination of the Euclidean distance and the angular distance intervening between the points. Namely, we use Euclidean metrics to compute within-domain distance, while for dimensions from different domains we use the Manhattan distance metrics, as suggested in [11,15]. Weights assigned to domain dimensions are affected by the context, too, so the resulting weighted Euclidean distance $dist_E$ is computed as follows

$$dist_E(p_1, p_2, k) = \sqrt{\sum_{i=1}^{n} w_i(p_{1,i} - p_{2,i})^2} \ ,$$

where i varies over the n domain dimensions, k is the context, and w_i are dimension weights.

The representation format adopted in conceptual spaces (e.g., for the concept *whale*) includes information such as:

```
02062744n,whale,dimension(x=350,y=350,z=2050),color(B=20,H=20,S=60),food=10
```

that is, the WordNet synset identifier, the *lemma* of the concept in the description, information about its typical dimensions, color (as the position of the instance on the three-dimensional axes of *brightness*, *hue* and *saturation*) and food. Of course, information about typical traits varies according to the species. Three domains with multiple dimensions have been defined:[7] *size*, *color* and *habitat*. Each quality in a domain is associated to a range of possible values. To avoid that larger ranges affect too much the distance, we have introduced a damping factor to reduce this effect; also, the relative strength of each domain can be parametrized.

We represent points as vectors (with as many dimensions as required by the considered domain), whose components correspond to the point coordinates, so that a natural metrics to compute the similarity between them is *cosine similarity*. Cosine similarity is computed as the cosine of the angle between the considered vectors: two vectors with same orientation have a cosine similarity 1, while two orthogonal vectors have cosine similarity 0. The normalized version of cosine similarity ($\hat{cs}$), also accounting for the above weights w_i and context k is computed as

$$\hat{cs}(p_1, p_2, k) = \frac{\sum_{i=1}^{n} w_i(p_{1,i} \times p_{2,i})}{\sqrt{\sum_{i=1}^{n} w_i(p_{1,i})^2} \times \sqrt{\sum_{i=1}^{n} w_i(p_{2,i})^2}}.$$

[7] We defined also further domains with one dimension (e.g., whiskers, wings, paws, fang, and so forth), but for our present concerns they are of less interest. The conceptual space is available at the URL `http://www.di.unito.it/~radicion/datasets/aic_13/conceptual_space.txt`.

Moreover, to satisfy the triangle inequality is a requirement upon distance in a metric space; unfortunately, cosine similarity does not satisfy triangle inequality, so we adopt a slightly different metrics, the *angular similarity* ($\hat{as}$), whose values vary over the range $[0, 1]$, and that is defined as

$$\hat{as}(p_1, p_2) = 1 - \frac{2 \cdot \cos^{-1} \cdot \hat{cs}(p_1, p_2, k)}{\pi}.$$

Angular distance allows us to compare the shape of animals disregarding their actual size: for example, it allows us to find that a python is similar to a viper even though it is much bigger.

In the metric space being defined, the distance d between individuals i_a, i_b is computed with the Manhattan distance, enriched with information about context k that indicates the set of weights associated to each domain. Additionally, the relevance of domains with fewer dimensions (that would obtain overly high weights) is counterbalanced by a normalizing factor (based on the work by [15]), so that such distance is computed as:

$$d(i_a, i_b, K) = \sum_{j=1}^{m} w_j \cdot \sqrt{|D_j|} \cdot dist_E\left(p_j(i_a), p_j(i_b), k_j\right) \ , \tag{1}$$

where K is the whole context, containing domain weights w_j and contexts k_j, and $|D_j|$ is the number of dimensions in each domain.

In this setting, the distance between each two concepts can be computed as the distance between two regions in a given domain, and then to combining them through the Formula 1. Also, we can compute the distance between any two region prototypes, or the minimal distance between their individuals, or we can apply more sophisticated algorithms: in all cases, we have designed a metric space and procedures that allow characterizing and comparing concepts herein. Although angular distance is currently applied to compute similarity in the *size* of the considered individuals, it can be generalized to further dimensions.

3 Experimentation

The evaluation consisted of an inferential task aimed at categorizing a set of linguistic descriptions. Such descriptions contain information related to concepts typical features. Some examples of these common-sense descriptions are: "the big carnivore with black and yellow stripes" denoting the concept of *tiger*, and "the sweet water fish that goes upstream" denoting the concept of *salmon*, and so on. A dataset of 27 "common-sense" linguistic descriptions was built, containing a list of stimuli and their corresponding category: this is the "prototypically correct" category, and in the following is referred to as the *expected* result.[8] The set of stimuli was devised by a team of neuropsychologists and philosophers in

[8] The full list is available at the URL `http://www.di.unito.it/~radicion/datasets/aic_13/stimuli_en.txt`.

Table 1: Results of the preliminary experimentation.

Test cases categorized	27	100.0%
[1.] Cases where $\mathcal{S}1$ and $\mathcal{S}2$ returned the same category	24	88.9%
[2a.] Cases where $\mathcal{S}1$ returned the expected category	25	92.6%
[2b.] Cases where $\mathcal{S}2$ returned the expected category	26	96.3%
Cases where $\mathcal{S}1$ OR $\mathcal{S}2$ returned the expected category	27	100.0%

the frame of a broader project, aimed at investigating the role of visual load in concepts involved in inferential and referential tasks. Such input was used for querying the system as in a typicality based question-answering task. In Information Retrieval such queries are known to belong to the class of "informational queries", i.e., queries where the user intends to obtain information regarding a specific information need. Since it is characterized by uncertain and/or incomplete information, this class of queries is by far the most common and complex to interpret, if compared to queries where users can search for the URL of a given site ('navigational queries'), or look for sites where some task can be performed, like buying music files ('transactional queries') [16].

We devised some metrics to assess the accuracy of the system, and namely we recorded the following information:

1. how often $\mathcal{S}1$ and $\mathcal{S}2$ returned in output the same category;
2. in case different outputs were returned, the accuracy obtained by $\mathcal{S}1$ and $\mathcal{S}2$:
   2a. the accuracy of $\mathcal{S}1$. This figure is intended to measure how often the top ranked category $\mathsf{c_0}$ returned by $\mathcal{S}1$ is the same as that expected.
   2b. the accuracy of $\mathcal{S}2$, that is the second category returned in the output pair $\langle \mathsf{c.}, \mathsf{cc} \rangle$. This figure is intended to measure how often the cc category is the appropriate one w.r.t. the expected result. We remark that cc has not been necessarily computed by starting from $\mathsf{c_0}$: in principle any $\mathsf{c_i} \in \mathsf{C}$ might have been used (see also Algorithm 1, lines 3 and 15).

The results obtained in this preliminary experimentation are presented in Table 1. All of the stimuli were categorized, although not all of them were correctly categorized. However, the system was able to correctly categorize a vast majority of the input descriptions: in most cases (92.6%) $\mathcal{S}1$ alone produces the correct output, with considerable saving in terms of computation time and resources. Conversely, none of the concepts (except for one) described with typical features would have been classified through classical ontological inference. It is in virtue of the former access to conceptual spaces that the whole system is able to categorize such descriptions. Let us consider, e.g., the description "The animal that eats bananas". The ontology encodes knowledge stating that *monkeys* are *omnivore*. However, since the information that *usually* monkeys eat bananas cannot be represented therein, the description would be consistent to all omnivores. The information returned would then be too informative w.r.t. the granularity of the expected answer.

Another interesting result was obtained for the input description "the big herbivore with antlers". In this case, the correct answer is the third element in the list C returned by $\mathcal{S}1$; but thanks to the categorization performed by $\mathcal{S}2$, it is returned in the final output pair (see Algorithm 1, line 8).

Finally, the system revealed to be able to categorize stimuli with typical, though ontologically incoherent, descriptions. As an example of such a case we will consider the categorization results obtained with the following stimulus: "The big fish that eats plankton". In this case the prototypical answer expected is *whale*. However, whales properly are mammals, not fishes. In our hybrid system, $\mathcal{S}1$ component returns *whale* by resorting to prototypical knowledge. If further details were added to the input description, the answer would have changed accordingly: in this sense the categorization performed by $\mathcal{S}1$ is non monotonic in nature. When then C (the output of $\mathcal{S}1$) is checked against the ontology as described by the Algorithm 1 at lines 7–13, and an inconsistency is detected,[9] the consistency of the second result in C (*shark* in this example) is tested against the ontology. Since this answer is an ontologically compliant categorization, then this solution is returned by the $\mathcal{S}2$ component. The final output of the categorization is then the pair $\langle \mathit{whale}, \mathit{shark} \rangle$: the first element, prototypically relevant for the query, would have not been provided by querying a classical ontological representation. Moreover, if the ontology recorded the information that also other fishes do eat plankton, the output of a classical ontological inference would have included them, too, thereby resulting in a too large set of results w.r.t. the intended answer.

4 Related work

In the context of a different field of application, a solution similar to the one adopted here has been proposed in [17]. The main difference with their proposal concerns the underlying assumption on which the integration between symbolic and sub-symbolic system is based. In our system the conceptual spaces and the classical component are integrated at the level of the representation of concepts, and such components are assumed to carry different –though complementary- conceptual information. On the other hand, the previous proposal is mainly used to interpret and ground raw data coming from sensor in a high level symbolic system through the mediation of conceptual spaces.

In other respects, our system is also akin to that ones developed in the field of the computational approach to the above mentioned dual process theories. A first example of such "dual based systems" is the *mReasoner* model [18], developed with the aim of providing a computational architecture of reasoning based on the mental models theory proposed by Philip Johnson-Laird [19]. The *mReasoner* architecture is based on three components: a system 0, a system 1 and a system 2. The last two systems correspond to those hypothesized by the dual process approach. System 0 operates at the level of linguistic pre-processing. It parses

[9] This follows by observing that $\mathsf{c}_0 = \mathit{whale}$, $\mathsf{cc} = \mathit{shark}$; and $\mathit{whale} \subset \mathit{mammal}$, while $\mathit{shark} \subset \mathit{fish}$; and *mammal* and *fish* are disjoint.

the premises of an argument by using natural language processing techniques, and it then creates an initial intensional model of them. System 1 uses this intensional representation to build an extensional model, and uses heuristics to provide rapid reasoning conclusions; finally, system 2 carries out more demanding processes to searches for alternative models, if the initial conclusion does not hold or if it is not satisfactory. Another system that is close to our present work has been proposed by [20]. The authors do not explicitly mention the dual process approach; however, they build a system for conversational agents (chatbots) where agents' background knowledge is represented using both a symbolic and a subsymbolic approach. They also associate different sorts of representation to different types of reasoning. Namely, deterministic reasoning is associated to symbolic (system 2) representations, and associative reasoning is accounted for by the subsymbolic (system 1) component. Differently from our system, however, the authors do not make any claim about the sequence of activation and the conciliation strategy of the two representational and reasoning processes. It is worth noting that other examples of this type of systems can be considered that are in some sense akin to the dual process proposal: for example, many hybrid, symbolic-connectionist systems –including cognitive architectures such as, for example, CLARION (`http://www.cogsci.rpi.edu/~rsun/clarion.html`)–, in which the connectionist component is used to model fast, associative processes, while the symbolic component is responsible for explicit, declarative computations (for a deeper discussion, please refer to [21]). However, at the best of our knowledge, our system is the only one that considers this hybridization with a granularity at the level of individual conceptual representations.

5 Conclusions and future work

In this paper we presented a cognitively inspired system to extend the representational and reasoning capabilities of classical ontological representations. We tested it in a pilot study concerning a categorization task involving typicality based queries. The results show that the proposed architecture effectively extends the reasoning and representational capabilities of formal ontologies towards the domain of prototype theory.

Next steps will be to complete the implementation of current system: first, we will work to the automatization of the Information Extraction from linguistic descriptions, and then to the automatization of the mapping of the extracted information onto the conceptual representations in $\mathcal{S}1$ and $\mathcal{S}2$. In near future we will also extend the coverage of the implemented system to further domains.

Yet, we are designing a learning setting to modify weights in conceptual spaces according to experience (thereby qualifying the whole system as a supervised learning one). This line of research will require the contribution of theoretical and experimental psychologists, to provide insightful input to the development of the system, and experimental corroboration to its evolving facets, as well. Future work will also include the evaluation of the system on web data, namely to experiment by using search engine web logs, in order to verify whether

and to what extent the implemented system matches the actual users' informational needs.

References

1. Frixione, M., Lieto, A.: Dealing with Concepts: from Cognitive Psychology to Knowledge Representation. Frontiers of Psychological and Behavioural Science **2**(3) (July 2013) 96–106
2. Rosch, E.: Cognitive representations of semantic categories. Journal of experimental psychology: General **104**(3) (1975) 192–233
3. Minsky, M.: A framework for representing knowledge. In Winston, P., ed.: The Psychology of Computer Vision. McGraw-Hill, New York (1975) 211–277
4. Brachman, R.J., Levesque, H.J.: Readings in Knowledge Representation. Morgan Kaufmann Pub (1985)
5. Brachmann, R.J., Schmolze, J.G.: An overview of the KL-ONE knowledge representation system. Cognitive Science **9**(2) (April 1985) 171–202
6. Frixione, M., Lieto, A.: The computational representation of concepts in formal ontologies-some general considerations. In: KEOD. (2010) 396–403
7. Stanovich, K.E., West, R.F.: Individual differences in reasoning: Implications for the rationality debate? Behavioral and brain sciences **23**(5) (2000) 645–665
8. Evans, J.S.B., Frankish, K.E.: In two minds: Dual processes and beyond. Oxford University Press (2009)
9. Kahneman, D.: Thinking, fast and slow. Macmillan (2011)
10. Machery, E.: Doing without concepts. Oxford University Press Oxford (2009)
11. Gärdenfors, P.: Conceptual Spaces. MIT Press (2000)
12. Frixione, M., Lieto, A.: Representing concepts in formal ontologies: Compositionality vs. typicality effects. Logic and Logical Philosophy **21**(4) (2012) 391–414
13. Frixione, M., Lieto, A.: Representing Non Classical Concepts in Formal Ontologies: Prototypes and Exemplars. In: New Challenges in Distributed Information Filtering and Retrieval. Volume 439 of Studies in Computational Intelligence. (2013) 171–182
14. Smith, E.R., Branscombe, N.R.: Category accessibility as implicit memory. Journal of Experimental Social Psychology **24**(6) (1988) 490–504
15. Adams, B., Raubal, M.: A metric conceptual space algebra. In Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G., eds.: COSIT. Volume 5756 of Lecture Notes in Computer Science., Springer (2009) 51–68
16. Jansen, B.J., Booth, D.L., Spink, A.: Determining the informational, navigational, and transactional intent of web queries. Information Processing & Management **44**(3) (2008) 1251–1266
17. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for artificial vision. Artificial Intelligence **89**(1–2) (1997) 73 – 111
18. Khemlani, S., Johnson-Laird, P.: The processes of inference. Argument & Computation **4**(1) (2013) 4–20
19. Johnson-Laird, P.: Mental models in cognitive science. Cognitive Science **4**(1) (1980) 71–115
20. Pilato, G., Augello, A., Gaglio, S.: A modular system oriented to the design of versatile knowledge bases for chatbots. ISRN Artificial Intelligence **2012** (2012)
21. Frixione, M., Lieto, A.: Formal Ontologies and Semantic Technologies: A Dual Process Proposal for Concept Representation. Philosophia Scientiae (forthcoming)

Introducing Sensory-motor Apparatus in Neuropsychological Modelization

Onofrio Gigliotta[1], Paolo Bartolomeo[2,3], and Orazio Miglino[1]

[1] University of Naples Federico II, Naples, Italy
onofrio.gigliotta@unina.it orazio.miglino@unina.it
[2] Centre de Recherche de l'Institut du Cerveau et de la Moelle èpiniére, Inserm U975, UPMC-Paris6, Paris, France
paolo.bartolomeo@gmail.com
[3] Department of Psychology, Catholic University, Milan, Italy

Abstract. Mainstream modeling of neuropsychological phenomena has mainly been focused to reproduce their neural substrate whereas sensory-motor contingencies have attracted less attention. In this study we trained artificial embodied neural agents equipped with a pan/tilt camera, provided with different neural and motor capabilities, to solve a well known neuropsychological test: the cancellation task. Results showed that embodied agents provided with additional motor capabilities (a zooming motor) outperformed simple pan/tilt agents, even those equipped with more complex neural controllers. We concluded that the sole neural computational power cannot explain the (artificial) cognition which emerged throughout the adaptive process.

Keywords: Neural agents, Active Vision, Sensory motor integration, Cancellation task

1 Introduction

Mainstream models of neuropsychological phenomena are mainly based on artificial bioinspired neural networks that explain the neural dynamics underlying some neurocognitive functions (see for example [4]). Much less attention has been paid to modeling the structures that allow individuals to interact with their environment, such as the sensory-motor apparatus (see [5] for an exception). The neurally-based approach is based on the assumption that the neural computational power and its organization is the main source of the *mental life*. Alternatively, as stated by eminent theorists [8, 9, 11], cognition could be viewed as a process that emerges from the interplay between environmental requests and organisms' resources (i.e. neural computational power, sensory-motor apparatus, body features, etc.). In other words, cognition comes from the adaptive history (phylogenetic and/or ontogenetic) in which all living organisms are immersed and take part. This theoretical perspective leads to building up artificial models that take into account, in embryonic form, neural structures, sensory-motor apparatus, environment structure and adaptation processes (phylogenetic and/or

ontogenetic). This modelization approach is developed by the interdisciplinary field of Artifcial Life and it is widely used in order to modelize a large spectrum of natural phenomena[3, 10, 6, 7]. In this study we applied artificial life techniques to building up neural-agents able to perform a well known neuropyschological task, the cancellation task, currently used to study the neurocognitive functions related to spatial cognition. Basically, this task is a form of visual search and it is considered as a benchmark to detect spatially-based cognitive deficits such as visual neglect [1].

2 Materials and Methods

2.1 The cancellation task

The cancellation task is a well known diagnostic test used to detect neuropsychological deficits in human beings. The test material typically consists of a rectangular white sheet which contains randomly scattered visual stimuli. Stimuli may be of two (or more) categories (for example triangles and squares, lines and dots, A and C letters, etc.). Figure 1a shows an example of the task. Subjects are asked to find and cancel by a pen stroke all the items of a given category (e.g. *open circles*). Fundamentally, it is a visual search task where some items are coded as distractors and other represent targets (the items to cancel). Brain-damaged patients can fail to cancel targets in a sector of space, typically the left half of the sheet after a lesion in the right hemisphere (visual neglect, see figure 1b).Here we simulated this task through a virtual sheet (a bitmap) in which a set of targets and distractors are randomly drawn (Fig. 1c), and trained neural agents provided with a specific sensory-motor apparatus, described in the next section, to perform the task.

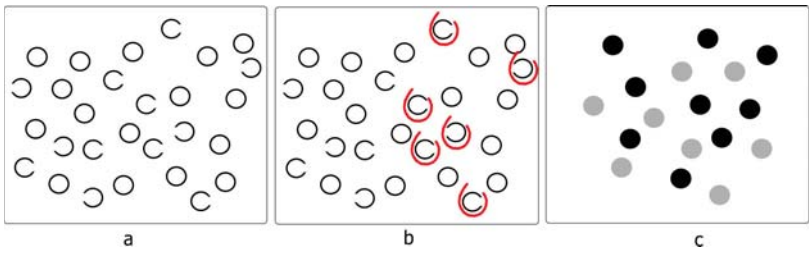

Fig. 1. a) Cancellation task in which targets are open circles and full circles are distractors; b) open circles canceled with a circular mark; c) cancellation task implemented in our experiments: grey filled circles are targets and black ones distractors

2.2 The neural agent's sensory-motor apparatus

A neural agent is equipped with a pan/tilt camera provided with a motorized zoom and an actuator able to trigger the cancellation behavior (Fig.2). The camera has a resolution of 350x350 pixels. Two motors allow the camera to explore the visual scene by controlling rotation around x and y axes while a third motor controls the magnification of the observed scene. Finally, the fourth actuator triggers a cancellation movement that reproduces in a simplified fashion the behavior shown by human individuals when asked to solve the task. The

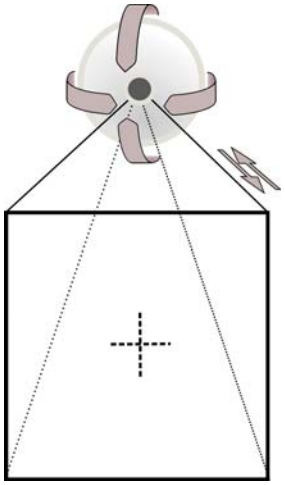

Fig. 2. The sensory-motor apparatus: two motors control rotation around two axes, one motor controls the zoom and a supplementary motor (not depicted) triggers the cancellation behaviour.

behavior of the neural agents is controlled by a neural network able to control the four actuators and to manage the camera visual input. The camera output does not gather all the pixel data, but pre-processes visual information using an artificial retina made up of 49 receptors (Fig. 3, right). Visual receptors are equally distributed on the surface of the camera; each receptor has a round visual field with a radius of 25 pixel. The activation of each receptor is computed by averaging the luminance value of the perceived scene (Fig. 3, left)

2.3 The cancellation task on the artificial neural agent

In order to simulate a form of cancellation task in silico, we trained neural agents endowed with different neural architectures to perform the cancellation task. In particular, we presented a set of randomly scattered stimuli made up of

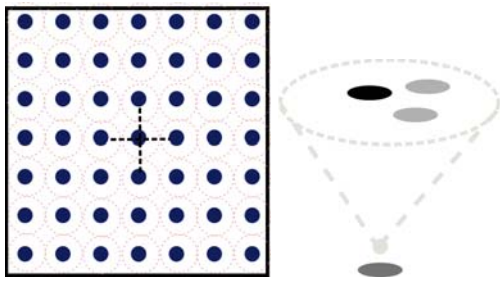

Fig. 3. Right. Neural agent's retina. Receptors are depicted as blue filled circle, receptive fields as dotted red circles. Left. Receptor activation is computed averaging the luminance value of the perceived stimuli.

distractors (black stimuli) and targets (grey stimuli) (Fig. 4) and rewarded neural agents for the ability to find (by putting the center of their retina over a target stimulus) and cancel/mark correct stimuli (activating the proper actuator).

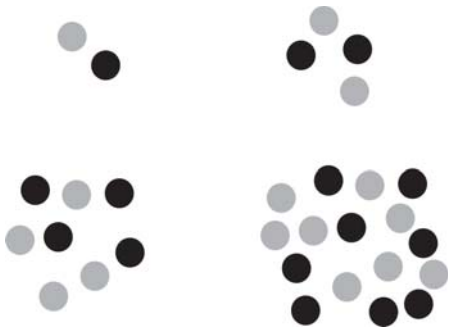

Fig. 4. Random patterns of targets (gray filled circles) and distractors (black filled circles)

2.4 Experiments

In order to perform the cancellation task, an agent has to develop (1) the ability to search for stimuli, and (2) to decide whether a stimulus is a target or not. To study how these abilities emerge we used controllers which were able to learn and self-adapt to perform the task. We provided agents with neural networks with different architectures designed by varying the number of internal neurons, the pattern of connections and the motor capabilities. In particular, we designed four architectures of increasing complexity (Fig. 5). Complexity was determined

first by the number of neurons and by their connections. In this case more complexity turns on more computational power that a controller can manage. Second, complexity can be related to the body in terms of sensory or motor resources that can be exploited to solve a particular task.

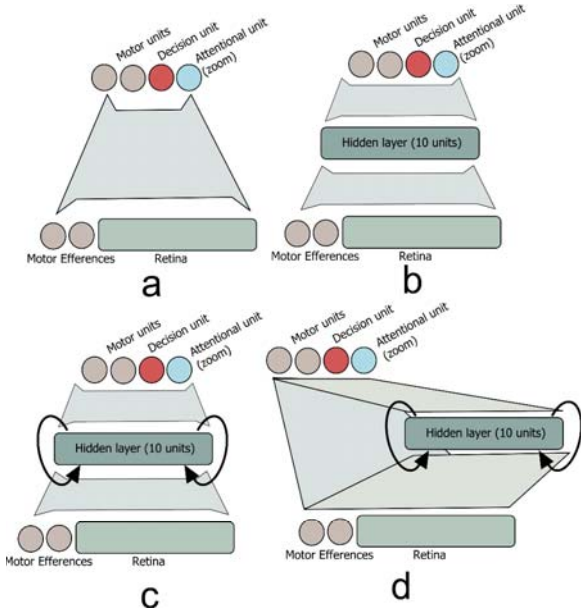

Fig. 5. Networks trained for the cancellation task: a) Perceptron; b)Feed forward neural networks with a 10 neurons hidden layer; c) Network b with a recurrent connection; d) Network c with a direct input-output connection layer.

In 8 evolutionary experiments, we trained neural agents by varying the controllers' architecture (4 conditions) and by adding the possibility to use or not the zooming actuator (2 conditions). For each experiment 10 populations of artificial agents were trained through a standard genetic algorithm [8] for 1000 generations. For each generation neural agents were tested 20 times with random patterns of target and distractor stimuli. Each agent was rewarded for its ability to explore the visual scene and correctly cancel/mark target stimuli.

3 Results

For each evolutionary experiment we post-evaluated the best ten individuals for the ability to correctly mark target stimuli. In particular, we tested each individual with 800 different random stimuli patterns.The rationale behind the post

evaluation is twofold. First, during evolution each agent experienced a small number of possible visual patterns (20); second, the reward function was made up of two parts so as to avoid bootstrapping problems: one component to reward exploration and the second one to reward correct cancellations. Results are reported as proportion correct in cancellation tests. Figure 6 reports the post-evaluation results for each architecture in each motor condition: with the ability to operate the zoom (Fig. 6 a,b,c and d) and without this ability (a-, b-, c- and d-).

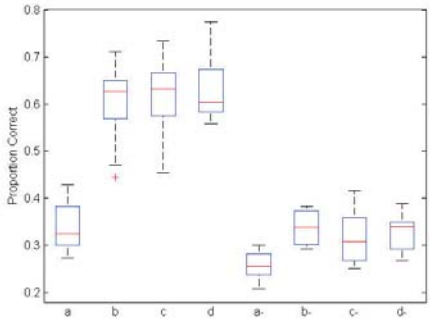

Fig. 6. Boxplots containing the post evaluation performance for each evolutionary experiment. Each boxplot reports the performance of the best 10 individuals.

For all the neural networks we found significant differences ($p<0.001$, two-tailed Mann-Whitney U test) between the condition presence/absence of the capacity to zoom incoming stimuli. In both groups there were significant differences between network a and the remaining networks, but no significant difference emerged between networks b, c and d. Interestingly, there were no significant differences between a, $b-$, $c-$, and $d-$. This last result suggests that a greater computational power can replace to some extent the absence of a zooming capacity. As mentioned above, neglect patients fail to process information coming from the left side of space. However healthy individuals can also show mild signs of spatial bias in the opposite direction (i.e., penalizing the right side of space), a phenomenon termed pseudoneglect [12]. In order to asses if such bias could simply have emerged as a side effect of the training process, we tested the best evolved individuals of the network d with a set of 200 couples of target stimuli placed symmetrically respect to the x axes of the artificial agent. Results (Fig. 7) show that only one individual (nr. 3 in Fig. 7) did not present a significant left-right difference, while all the remaining had different degrees of spatial preference.

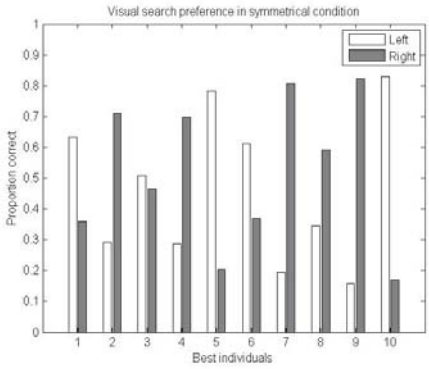

Fig. 7. Individual proportion correct in the selection of left or right-sided targets as first visited item.

4 Conclusion

At variance with the mainstream approach in the modeling of neuropsychological phenomena, mainly focused on reproduction of the neural underpinnings of cognitive mechanisms, we showed that having a proper motor actuator can greatly improve the performance of evolved neural agents in a cancellation task. In particular, we demonstrated that an appropriate motor actuator (able to implement a sort of attentional/zooming mechanism) can overcome the limits associated with intrinsic computational power (e.g. number of internal neurons and neural connections in our case). Second, we showed that spatial bias in stimulus selection in *healthy neural agents* can be a side effect of the training process. In future extensions of this work we plan to test *injured neural agents*, evaluate biologically-inspired neural architectures following recent research results on brain attentional networks[2] and to extend the range of different explored sensory-motor capabilities.

References

1. P Azouvi, C Samuel, A Louis-Dreyfus, T Bernati, P Bartolomeo, J-M Beis, S Chokron, M Leclercq, F Marchal, Y Martin, G de Montety, S Olivier, D Perennou, P Pradat-Diehl, C Prairial, G Rode, E Siroff, L Wiart, and M Rousseaux. Sensitivity of clinical and behavioural tests of spatial neglect after right hemisphere stroke. *Journal of Neurology, Neurosurgery & Psychiatry*, 73(2):160–166, 2002.
2. Paolo Bartolomeo, Michel Thiebaut de Schotten, and Fabrizio Doricchi. Left unilateral neglect as a disconnection syndrome. *Cerebral Cortex*, 17(11):2479–2490, 2007.
3. M. Bedau. Artificial life: organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*, 7(11):505–512, November 2003.

4. Marco Casarotti, Matteo Lisi, Carlo Umilta, and Marco Zorzi. Paying attention through eye movements: A computational investigation of the premotor theory of spatial attention. *J. Cognitive Neuroscience*, 24(7):1519–1531, 2012.
5. Andrea Di Ferdinando, Domenico Parisi, and Paolo Bartolomeo. Modeling orienting behavior and its disorders with ecological neural networks. *Journal of Cognitive Neuroscience*, 19(6):1033–1049, 2007.
6. Onofrio Gigliotta and Stefano Nolfi. Formation of spatial representations in evolving autonomous robots. In *Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007)*, pages 171–178, Piscataway, NJ, 2007. IEEE Press,.
7. Onofrio Gigliotta, Giovanni Pezzulo, and Stefano Nolfi. Evolution of a predictive internal model in an embodied and situated agent. *Theory in Biosciences*, 130(4):259–276, 2011.
8. Stefano Nolfi and Dario Floreano. *Evolutionary Robotics*. Mit Press, 2000.
9. Rolf Pfeifer and Josh Bongard. *How the body shapes the way we think*. The Mit Press, 2006.
10. Michela Ponticorvo and Orazio Miglino. Encoding geometric and non-geometric information: a study with evolved agents. *Animal Cognition*, 13(1):157–174, 2010.
11. Robert F. Port and Timothy Van Gelder. *Mind as motion : explorations in the dynamics of cognition*. Bradford Book, Cambridge (Mass.), London, 1995. A Bradford books.
12. W Vingiano. Pseudoneglect on a cancellation task. *International Journal of Neuroscience*, 58(1-2):63–67, 1991.

How Affordances can Rule the (Computational) World

Alice Ruggeri and Luigi Di Caro

Department of Computer Science, University of Turin
Corso Svizzera 185, Torino, Italy
{ruggeri,dicaro}@di.unito.it

Abstract. In this paper we present an ontology representation which models the reality as not objective nor subjective. Relying on a Gibsonian vision of the world to represent, our assumption is that objects naturally give suggestions on how they can be used. From an ontological point of view, this leads to the problem of having different representations of identical objects depending on the context and the involved agents, creating a more realistic multi-dimensional object space to be formally defined. While avoiding to represent purely subjective views, the main issue that needs to be faced is how to manage the highest complexity with the minimum resource requirements. More in detail, we extend the idea of ontologies taking into account the subjectivity of the agents that are involved in the interaction. Instead of duplicating objects, according to the interaction, the ontology changes its aspect, fitting the specific situations that take place. We propose the centerpieces of the idea as well as suggestions of applications that such approach can have in several domains, ranging from Natural Language Processing techniques and Ontology Alignment to User Modeling and Social Networks.

1 Introduction and Research Questions

We usually refer to the term *ontology* with several meanings in mind. Generally speaking, it can be defined as an attempt to represent the world (or a part of it) in an objective way. This is usually reflected in a representation of objects with fixed properties, independently from the interaction schemes. From the other side, there can be a purely subjective vision that every single agent may have. Our idea regards an ontological modeling of the behavior of intelligent agents, built on top of the concept of affordance introduced by [1] to describe the process underlying the perception. Generally speaking, Gibson claimed that objects assume different meanings depending on the context, and more specifically, according to which animal species interacts with them. The verb "*to afford*", in fact, implies the complementarity of the animal with the environment. In this sense, it is a distributed property between the agent, the action, and the object (i.e., the one that receives the action). All these components contribute to the meaning of the whole situation. An important characteristic of an affordance is that it is not objective nor subjective: actually, it cuts across the dichotomy

between objective and subjective. More in detail, it relies on both environmental and behavioral facts, turning in both directions: from the environment point of view and the observer's one. Still, an interesting Gibsonian point of analysis is that the body depends on its environment, but the existence of the latter does not depend on the body. At this point, we recall to the classic dichotomy of the two main types of knowledge: explicit (to know what) and implicit (to know how) [2]. As an example, let us consider a surface. A table can offer an affordance of walking to a fly but not to an elephant, due to their different sizes and weights with respect to its surface. Different species can perform different actions on the same object but also the same action can be performed differently by the two species. Let us now consider an apple: it can be eaten by a worm living inside it, while an elephant can chew it. This situation cannot be modeled in a hypothetically objective approach to ontologies, whereas, according to a subjective approach, it would result in a multiplicity of separated ontologies. The problem of having such a large and fine-grained object space is that every single species has to be duplicated for each pair of species/agent, conducting to misalignments and relative problematic management. However, the purpose of a computational ontology is not to specify what "exists" and what "does not exist", but to create a specific knowledge base, which is an artifact of men, containing concepts related to the domain of investigation and that it will be used to perform certain types of computation. In our view, according to the interaction, the ontology should change its aspect fitting the specific situations that the ontologists would want to represent. From this, some questions arise:

- How to change the primitive of ontology representation in order to take into account affordances?
- What kind of direct applications may be found, and how can they be implemented?

However, Gibson limits his approach only to objects, whereas we aim at considering also technological artifacts and institutional entities from the socially constructed reality, like schools, organizations, and so forth. The aim of this paper is not purely theoretical, since we want to apply the idea in several domains of Computer Science, from natural language understanding to user modelling.

With the introduction of an affordance level, we increase the flexibility of the world we are going to represent. More specifically, with an augmented representation of the interaction between agents and objects, we start representing the tacit and implicit knowledge to model the explicit one.

2 Cognitive-based Computational Ontologies

The idea of going towards cognitive approaches for the construction of ontologies has been already proposed in [3, 4]. Our starting point is to compare approaches to ontologies that represent purely objective rather than subjective views of the world. On the one hand, in the objective view, all objects have the same features and belong to fixed classes. The actions that can be performed on the objects

are the same and have the same meaning regardless of the agent performing the action. On the other hand, in a purely subjective scenario, we have a plurality of possibly inconsistent ontologies, one for each agent or species. Besides being too broad and complex to represent, the main problem would be that the same concept would be unrelated to the corresponding ones in the ontologies of other agents. This leads to disalignments, to the impossibility to reuse part of the representations even if the concepts are similar, and to difficulties in maintaining the knowledge base.

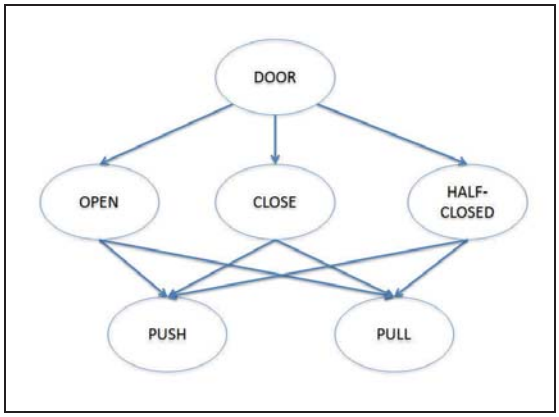

Fig. 1. The purely objective view of the world. The door can have different states like open, close (and other ones in the middle, like half-close). Then, there exist different actions that may change its status, like "push" and "pull".

To represent these issues, our starting point is to use formal ontologies. In general, formal ontologies are inspired to the basic principles of the First Order Logic [5], where the world is explained by the existence of defined objects and fixed relationships among them. This belongs to a physical and static view of the world. Figure 1 shows how this representation reduces to the existence of many objects and different behaviors associated with them. The same actions are offered to all agents interacting with the object, independently of the properties of these agents.

Let us now consider the action of opening a door, first performed by a person and then from a cat. In this case, depending on the subject and its physical capabilities, the action of opening the door is performed in different modalities. From our knowledge, we are able to distinguish a human from a cat from many things; for example, the human has fingers and hands. For this reason, we can easily imagine that such action will be completed by the use of a door handle. Switching the subject "person" with "cat", the action will be mentally visualized

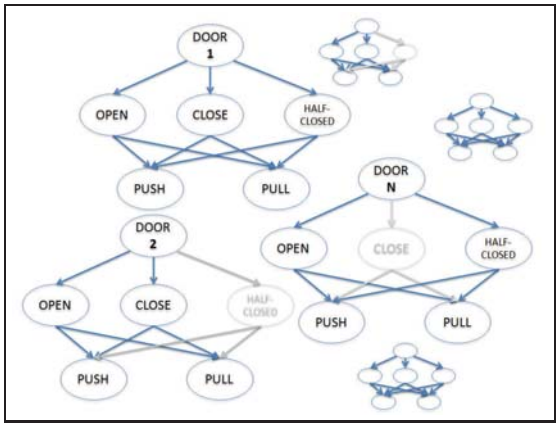

Fig. 2. The purely subjective view of the world. The enumeration of all instances and relationships without any ontology alignments produce a huge object space that turns out to be impossible to treat computationally.

in a different shape. The cat does not have fingers and it usually does not use any door handle[1]. This dependency between object and subject influences several activities: the mental image of the action by the subject or by another agent figuring out the situation, and the interpretation of a sentence describing it; then, in Computer Science scenarios, the implementation of the action on the object must be made differently depending on the subject interacting with the system. A completely subjective vision of how a situation can be would lead to an excessive chaos and a huge proliferation of instances, classes and relationships, as illustrated in Figure 2.

Our hypothesis is illustrated in Figure 3: we introduce concepts which have different perspectives depending on the kind of agent or species is interacting with them. Instead of having an object duplicated in different classes according to the different possible behaviors afforded to different agents (which would be reflected in an ontology with countless disjoint subclasses of the same object), we now have more inner classes depending on the agent who performs the action. The door provides two different ways to interact with it (the set of methods, if we want to use a programming language terminology): a way for a *human* user and on the other side the one for a *cat*. These two ways have some common actions with different ways to be performed (implementations), but they can also offer additional actions to their agents or players. For example a human can also lock a door with the key or shut it, while a cat can not. For example, the behavioral consequence of "how to interact with the door" can be "opened by the handle" rather than "pushed leaning on it", and the way the action will

[1] Someone may argue with that.

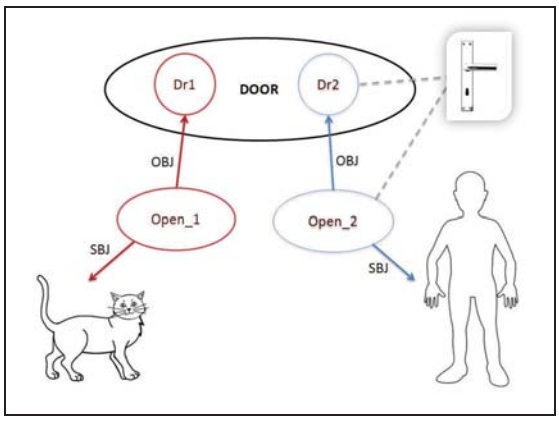

Fig. 3. From an ontological point of view, when the subject takes part to the meaning of the situation under definition, there is no need of concept duplication. Instead, the ontology has to provide mechanisms to contextualize the relationships according to the subject, eventually with the addition of specific properties or objects (as the door handle for the human subject).

be performed is determined by who is the subject of the action. The second example has a different character, since it refers to a technological artifact, i.e., a printer. As such, the object can have more complex behaviours and above all the behaviours do not depend only on the physical properties of the agents interacting with it but also with other properties, like the role they play and thus the authorizations they have. The printer provides two different roles to interact with it (the set of methods): the role of a *normal user*, and a role of *super user*. The two roles have some common methods (roles are classes) with different implementations, but they also offer other different methods to their agents. For example, normal users can print their documents and the number of printable pages is limited to a maximum determined (the number of pages is counted, and this is a role attribute associated with the agent). Each user must be associated with a different state of the interaction (the role has an instance with a state). Super users have the printing method with the same signature, but with a different implementation: they can print any number of pages; furthermore, they can reset the page counter (a role can access the status of another role, and, therefore, the roles coordinate the interaction). Note that the printer has also different properties for different roles and not only behaviours: for a normal user there is a number of remaining copies, for a super user that number is always infinite. A classical ontological view of the printer case is shown in Figure 4, while Figure 5 shows an example of how an intelligent system like a printer works depending on who is the user performing the action. The printer is divided into different "inner classes" (using a programming language terminology), depending

on how many number of remaining copies are printable (marked as *nc* within the figure). The third example we consider is of a totally different kind. There is no more physical object, since the artifact is an institution, i.e., an object of the socially constructed reality [6]. Consider a university, where each person can have different roles like professor, student, guardian, and so forth. Each one of these will be associated to different behaviours and properties: the professors teach courses and give marks, have an income; the students give exams, have an id number, and so forth. Here the behaviour does not depend anymore on the physical properties but on the social role of the agent.

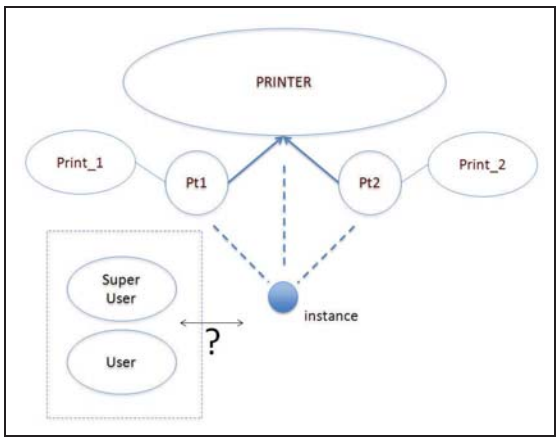

Fig. 4. A classic ontological view of the printer scenario. An instance has to belong to one of the three classes, but none of them captures the semantics associated to the interaction with the users. In case the new instance belongs to both printer pt1 and printer pt2, then it inherits all their methods, thus avoiding the differentiation at user-level.

The role of super user can safely access the state of other users and roles only if encapsulated in the printer. Hence the definition of the role should be given by the same programmer that defines the establishment (the class of the role belongs to the same class namespace, or, in Java terminology, it is included in that). In order to interact as user or super user, a particular behaviour is required. For example, in order to have the role of user, the user must have a certain type of account.

3 Applications

When we think at an object, what we perceive is not its qualities; rather, we get the affordances that it offers to the external world, in which the quality inhabits.

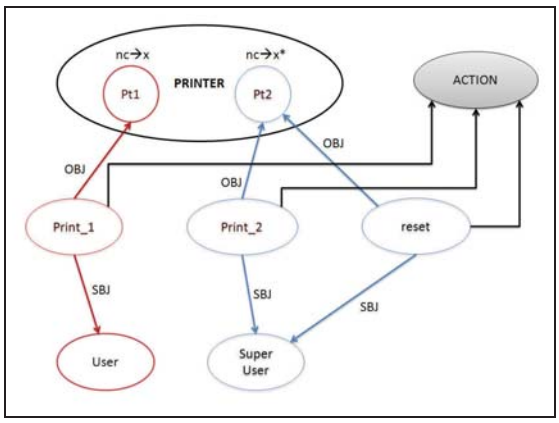

Fig. 5. An intelligent system where a printer works differently depending on who is the user performing the action.

Moreover, objects can be manufactured as well as manipulated. Some of them are transportable while others not; depending on the physical characteristics of an object, agents may perform distinct actions. In spite of this, however, it is not necessary (and possible) to distinguish all the features of an object. Perception combines the geometry of the world with behavioral goals and costs associated to them [7]. Still, positive and negative affordances are properties of things in reference to an observer, but not ownership of the experiences of the observer. [8] stated that all things, within themselves, have an enquiring nature that tell us what to do with them. In the end, we should not think about the existence or not of real things, but if the information is available to be perceived. If the information is not captured, the result is a misperception that may avoid the need of a tentative representation.

3.1 User Modeling

We discuss now the problem of modeling the ontology of different types of users and the ways they can interact one to each other. We can find a link between the User Modeling and the ontological theory of Von Uexküll [9], which can be expressed as follows: there is a circle which is a functional model of the agent who performs the action in its environment. The object of the action acquires a meaning if the action is implemented, thus through the concept of interaction. Von Uexküll theorized that each living organism was surrounded by a neighborhood perceived in a subjective manner, which he called umwelt. The environment is formed not by a single entity thet relates in the same way all living beings, but as an entity that changes its appearance depending on the species that perceives it. He reports, for example, the case of a “forest” that is seen differently

from the hypothetical eyes of a forest (as a set of trees to be treated and cut), an agronomist (as an area to be tilled to make room for crops), or a child (as a magical place populated by strange creatures). Thus, affordances can be employed to fragment the subjective views of the same ontological concepts, related to users within a community. Instead of having multiple ontologies (with eventually minimal differences), there can be a single one together with some formally defined middle-layer interface that can entail the specificity of the users. For example, let us consider an ontology about beverages. If we take the concept "wine", it can be viewed under different perspective depending on the subjectivity of a wine expert rather than a wine consumer. The former may consider technical facets like taste, appearance and body that a standard wine consumer could not even have in mind.

3.2 Natural Language Processing

The concept of affordances can meet well-known tasks belonging to Computational Linguistics. In fact, if we consider the objects / agents / actions to be terms in text sentences, we can try to extract their meaning and semantic constraints by using the idea of affordances. For instance, let us think to the sentence "The squirrel climbs the tree". In this case, we need to know what kind of subject 'squirrel' is to figure out (and visually imagine) how the action will be performed. According to this, no particular issues come out from the reading of this sentence. Let us now consider the sentence "The elephant climbs the tree". Even if the grammatical structure of the sentence is the same as before, the agent of the action is different, and it obviously creates some semantic problems. In fact, from this case, some constraints arise; in order to climb a tree, the subject needs to fit to our mental model of "something that can climb a tree". In addition, this also depends on the mental model of "tree". Moreover, different agents can be both correct subjects of an action whilst they may produce different meanings in terms of how the action will be mentally performed. Consider the sentences "The cat opens the door" and "The man opens the door". In both cases, some implicit knowledge suggests the manner the action is done: while in the second case we may think at the cat that opens the door leaning to it, in the case of the man we probably imagine the use of a door handle. A study of these language dynamics can be of help for many NLP tasks like Part-Of-Speech tagging as well as more complex operations like dependency parsing and semantic relations extraction. Some of these concepts are latently studied in different disciplines related to statistics. Distributional Semantics (DS) [10] represents a class of statistical and linguistic analysis of text corpora that try to estimate the validity of connections between subjects, verbs, and objects by means of statistical sources of significance.

3.3 Social Networks

Social networks are a modern way people use to communicate and share information in general. Facebook[2], Twitter[3], Flickr[4] and others represent platforms to exchange personal data like opinions, pictures, thoughts on world wide facts, and related information. All these communities rely on the concept of user profile. A user profile is generally a set of personal information that regard the user in itself as well his activity within the community. Understanding the reference prototype of a user is central for many operations like information recommendation, user-aware information retrieval, and User Modeling-related tasks in general. In this context, the concept of affordance can be used in several scenarios. First, it can be a way to personalize the content to show to the user according to his interests and activity. This is massively done in today's web portals, where advertising is more and more adapted to the web consumers. Secondly, the whole content shared by 'user friends' can be filtered according to his profile, in the same way as in the advertising case. Notice that this does not have to do with privacy issues. In fact, a user may be not interested in all facts and activities coming from all his friends. Future social networking web sites may take into consideration such kind of personalization at user-context level.

3.4 Ontology Alignment

Ontology alignment, also called ontology matching, is the task of finding connections between concepts belonging to different ontologies. This is an important issue since usually identical domains are defined by using hand-crafted ontologies that differ in terms of vocabulary, granularity, and focus. [11] represents one of the most complete survey on the existing approaches. The concept of affordance can be thought as the conceptual bridge between the definition of a domain and the domain itself. In fact, the former is a view of the domain that takes into account the subjectivity and the context the concepts would fit with. Focusing on how to formalize such middle level can put the basis for a semantic-based ontology alignment that dodges most of the existing statistical techniques and their relative semantic blindness.

4 Related Work

In this section, we review the main works that are related to our contribution. For an exhaustive reading, it is worth to mention the ideas presented in [12–14] about the design of ontologies in Information Systems.

Mental models have been introduced by Johnson Laird [15], as an attempt to symbolic representations of knowledge to make it computable, i.e., executable by a computer. This concept is the basis of the most important human-computer

[2] https://www.facebook.com/
[3] https://twitter.com/
[4] http://www.flickr.com/

cognitive metaphor. A mental model is composed by tokens (elements) and relations which represent a specific state of things, structured in an appropriate manner to the processes that will have to operate on them. There is no a single mental model to which the answer is right and that corresponds to a certain state of things: a single statement can legitimately correspond to several models, although it is likely that one of these matches in the best way to describe the state of affairs. This allows to represent both the intension that the extension of a concept, namely the characteristic properties of the state described; the management procedures of the model are used to define the extension of the same concept, that is, the set of all possible states that describe the concept. Figures 6 and 7 show the case of an airplane and the resulting mental models that we create according to different types of action: recognize it or travel with it. Indeed, the action changes the type of perception we have of an object and the action takes different meanings depending on the interaction with the subject that performs it.

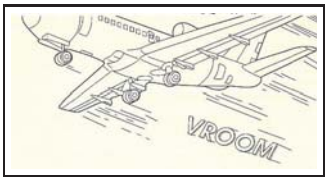

Fig. 6. A mental model of an airplane to recognize it. [15]

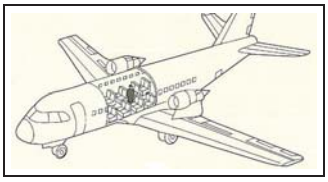

Fig. 7. A mental model of an airplane when travelling. [15]

From the mental models theory we then reach the mental images theory [16]. Mental images are not figures in a person's mind, but they are mental representations even in the absence of the corresponding visual stimuli. Unfortunately, the operation for defining how the images are built, formed, and transformed is still a controversial issue.

Another related work which can be considered as a starting point of our analysis is about the link between the Gestalt theory [17, 18] and the concept of

affordance in the original way introduced by Gibson for the perception of objects. Wertheimer, Kohler and Koffka, the founders of the Gestalt movement, applied concepts to perception in different modalities. In particular, it is important to remind the principle of complementarity between "figure" and "ground". In this paper we intend the ground as the contextual basis of an action; for instance, we can not understand the whole meaning(s) of a sentence if we do not consider the ground which surrounds the interaction. The perception process, as we know, is immediate; however, to understand a figure, the input must be somehow recognized and trasformed within our brain. The final output is then mediated by contextual and environmental facts: it is a dynamic and cooperative process. Another point that we want to focus on within this contribution is to create a connection between the Gestalt theory and the Natural Language Processing applications that we explained in previous sections. Again, let us think at the sentence "The cat opens the door". In this case, our basic knowledge of what the cat is and how it moves can be our ground or contextual layout; this is useful to understand the whole figure and to imagine how this action will be performed. In simple words, the Gestalt theory helps us say that the tacit knowledge about something (in this case, how the cat uses its paws) is shaped on the explicit knowledge of "what the door is". Following this perspective, the concepts are not analyzed in a dyadic way, but in a triadic manner, similarly to the Pierce's semiotic triangle of reference, which underlies the relationship between meaning, reference and symbols [19].

Then, in Object-Oriented programming, an inner class is a type of class defined as part of a top-level class, from which its existence depends. An inner class could even define a distinct concept with respect to the outer class, and this makes it different from being a subclass. Powerjava [20] is an extension of the Java language and a simple object-oriented language, where an objective and static view of its components is modified and replaced on the basis of the functional role that objects have inside. The behavior of a particular object is studied in relation to the interaction with a particular user. In fact, when we think at an object, we do it in terms of attributes and methods, referring to the interaction among the objects according to public methods and public attributes. The approach is to consider Powerjava roles as affordances, that is, instances that assume different identities taking into account the agents.

5 Conclusions and Future Work

In this paper we proposed a Gibsonian view to ontology representation; objects of a domain offer affordances that help the involved agents make the correct actions. This approach can have several applications in different domains. For instance, it can be used to model some natural language dynamics like the attachment among subjects, verbs and objects in textual sentences. From the ontological point of view, the concept of affordance can be seen as the different ways the same objects can be seen by different people with specific interests and characteristics. Still, User Modeling tasks like information recommendation may

be faced according to the definition of affordance. Social Networks like Facebook and Twitter play an important role in nowadays online information spreading, as they represent frameworks where subjective views of identical information come out naturally and from which it would be crucial some formal mechanisms of knowledge representation. To apply affordances to user-generated and shared data can be useful for a number of applications like user-aware content sharing, and targeted advertising. In future work, we aim at focusing on these applications in order to implement ways of building ontologies according to the concept of affordance while minimizing redundancy.

References

1. Gibson, J.: The concept of affordances. Perceiving, acting, and knowing (1977) 67–82
2. Dienes, Z., Perner, J.: A theory of implicit and explicit knowledge. Behavioral and Brain Sciences **22**(05) (1999) 735–808
3. Oltramari, A.: An introduction to hybrid semantics: the role of cognition in semantic resources. In: Modeling, Learning, and Processing of Text Technological Data Structures. Springer (2012) 97–109
4. Osborne, F., Ruggeri, A.: A prismatic cognitive layout for adapting ontologies. In: User Modeling, Adaptation, and Personalization. Springer (2013) 359–362
5. Smullyan, R.: First-order logic. Dover Publications (1995)
6. Searle, J.: Construction of social reality. Free Press (1995)
7. Proffitt, D.: Embodied perception and the economy of action. Perspectives on psychological science **1**(2) (2006) 110–122
8. Koffka, K.: Principles of gestalt psychology. (1935)
9. Von Uexküll, J.: Umwelt und innenwelt der tiere. Springer Berlin (1909)
10. Baroni, M., Lenci, A.: Distributional memory: A general framework for corpus-based semantics. Computational Linguistics **36**(4) (2010) 673–721
11. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The knowledge engineering review **18**(1) (2003) 1–31
12. Guarino, N.: Formal Onthology in Information Systems: Proceedings of the First International Conference (FIOS'98), June 6-8, Trento, Italy. Volume 46. IOS press (1998)
13. Sowa, J.F.: Conceptual structures: information processing in mind and machine. (1983)
14. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? International journal of human-computer studies **43**(5) (1995) 907–928
15. Johnson-Laird, P.: Mental models: Towards a cognitive science of language, inference, and consciousness. Number 6. Harvard University Press (1983)
16. Kosslyn, S.: Image and mind. Harvard University Press (1980)
17. Köhler, W.: Gestalt psychology. (1929)
18. Wertheimer, M., Köhler, W., Koffka, K.: Gestaltpsychologie. Einfuhrung in die neuere Psychologie. AW Zickfeldt, Osterwieck am Harz (1927)
19. Peirce, C.: Peirce on signs: Writings on semiotic by Charles Sanders Peirce. University of North Carolina Press (1991)
20. Baldoni, M., Boella, G., Van Der Torre, L.: powerjava: ontologically founded roles in object oriented programming languages. In: Proceedings of the 2006 ACM symposium on Applied computing, ACM (2006) 1414–1418

Latent Semantic Analysis as Method for Automatic Question Scoring

David Tobinski[1] and Oliver Kraft[2]

[1] Universität Duisburg Essen, Universitätsstraße 2, 45141 Essen
david.tobinski@uni-due.de,
WWW home page: www.kognitivismus.de
[2] Universität Duisburg Essen, Universitätsstraße 2, 45141 Essen

Abstract. Automatically scoring open questions in massively multiuser virtual courses is still an unsolved challenge. In most online platforms, the time consuming process of evaluating student answers is up to the instructor. Especially unexpressed semantic structures can be considered problematic for machines. Latent Semantic Analysis (LSA) is an attempt to solve this problem in the domain of information retrieval and can be seen as general attempt for representing semantic structure. This paper discusses the rating of one item taken from an exam using LSA. It is attempted to use documents in a corpus as assessment criteria and to project student answers as pseudo-documents into the semantic space. The result shows that as long as each document is sufficiently distinct from each other, it is possible to use LSA to rate open questions.

Keywords: Latent Semantic Analysis, LSA, automated scoring, open question evaluation

1 Introduction

Using software to evaluate open questions is still a challenge. Therefore, there are many types of multiple choice tests and short answer tasks. But there is no solution available in which students may train their ability to write answers to open questions, as it is required in written exams. Especially in online courses systems (like Moodle), it is up to the course instructor to validate open questions herself.

A common method to analyze text is to search for certain keywords, as it is done by simple document retrieval systems. This method can not take into account that different words may have the same or a similar meaning. In information retrieval this leads to the problem, that potentially interesting documents may not be found by a query with too few matching keywords. Latent Semantic Analysis (LSA, Landauer and Dumais 1997) faces this problem by taking the higher-order structure of a text into account. This method makes it possible to retrieve documents which are similar to a query, even if they have only a few keywords in common.

Considering this problem in information retrieval to score an open question seems to be a similar problem. Exam answers should contain important keywords, but contain their own semantic structure also. This paper attempts to rate a student's exam answer by using LSA. For that a small corpus based upon the accompanying book of the course "Pädagogische Psychologie" (Fritz et al. 2010) is manually created. It is expected that it is in general possible to rate questions this way. Further it is of interest what constraints have to be taken into account to apply LSA for question scoring.

2 Latent Semantic Analysis

LSA was described by Deerwester et al. 1990) as a statistical method for automatic document indexing and retrieval. Its advantage to other indexing techniques is that it creates a *latent* semantic space. Naive document retrieval methods search for keywords shared by a query and a corpus. They have the disadvantage that it is difficult or even impossible to find documents if the request and a potentially interesting document have a lack of shared keywords. Contrary to this, LSA finds similarities even if query and corpus have few words in common. Beside its application in the domain of Information Retrieval, LSA is used in other scientific domains and is discussed as theory of knowledge acquisition (1997).

LSA is based upon the Vector Space Model (VSM). This model treats a document and its terms as a vector in which each dimension of the vector represents an indexed word. Multiple documents are combined in a *document-term-matrix*, in which each column represents a document and rows represent a terms. Cells contain the term frequency of a document (Deerwester et al. 1990).

A matrix created this way may be weighted. There are two types of weighting functions. Local weighting is applied to a term i in document j and global weighting is the terms weighting in the corpus. $a_{ij} = local(i,j) * global(i)$, where a_{ij} addresses a cell of the document-term-matrix (Martin and Berry 2011). There are several global and local weight functions. Since Dumais attested LogEntropy to improve retrieval results better than other weight function (Dumais 1991), studies done by Pincombe (2004) or Jorge-Botana et al. (2010) achieved different results. Although there is no consensus about the best weighting, it has an important impact to retrieval results.

After considering the weighting of the document-term-matrix, Singular Value Decomposition (SVD) is applied. SVD decomposes a matrix X into the product of three matrices:

$$X = T_0 S_0 D_0^T \tag{1}$$

Component matrix T_0 contains the derived orthogonal term factors, D_0^T describes the document factors and S_0 contains singular values, so that their product recreates the original matrix X. By convention, the diagonal matrix S is arranged in descending order. This means, the lower the index of a cell, the more information is contained. By reducing S from m to k dimensions, the

product of all three matrices ($\widehat{X}$) is the best approximation of X with k dimensions. Choosing a good value for k is critical for later retrieval results. If too many dimensions remain in S, unnecessary information will stay in the semantic space. Choosing k too big will remove important information from the semantic space (Martin and Berry 2011).

Once SVD is applied and the reduction done, there are four common types of comparisons, where the first two comparisons are quite equal: (i) Comparing documents with documents is done by multiplying D with the square of S and transposition of D. The value of cell $a_{i,j}$ now contains the similarity of document i and document j in the corpus. (ii) The same method can be used to compare terms with terms. (iii) The similarity of a term and a document can be taken from the cells of $\widehat{X}$. (iv) For the purpose of information retrieval, it is important to find a document described by keywords. According to the VSM keywords are composed in a vector, which can be understood as a query (q). The following formula projects a query into semantic space. The result is called *pseudo-document* (D_q) (Deerwester et al. 1990):

$$D_q = q^T T S^{-1} \tag{2}$$

To compute similarity between documents and the pseudo-document, consine similarity is generally taken (Dumais 1991). In their studies Jorge-Botana et al. (2010) found out that Euclidean distance performs better than cosine similarity.

3 Application configuration

To verify if LSA is in general suitable for valuating open questions, students answers from psychology exam in summer semester 2010 are analyzed. The exam question requires to describe, how a text can be learned by using the three cognitive learning strategies memorization, organization and elaboration. Each correct description is rated with two points. A simple description is enough to answer the question correctly, it is not demanded to transfer knowledge by giving an example. For the evaluation brief assessment criteria are available, but due to the short length of the description of each criterion new criteria are created by using the accompanying book of the course as mentioned above.

For the assessment a corpus is created, where each document is interpreted as an assessment criterion, which is worth a certain number of points. This way quite small corpora are created. For example, if a question is worth four points the correlating corpus contains exact four documents and only a few hundred terms, sometimes even less. To reduce noise in the corpus a list of stopwords is used. Because the students answers are short in length, stemming is used in this application. Beside using stemming and a list of stopwords, the corpus is weighted. Pincombe (2004, 17) showed that for a small number of dimensions BinIDF weighting has a high correlation to human ratings. Since the number of dimensions is that low (see below) and a human rating is taken as basis for the evaluation of LSA in this application, the used corpus is weighted by BinIDF.

All calculations are done by using GNU R statistical processing language using "lsa"[3] library provided by CRAN. The library is based upon SVDLIBC[4] by Doug Rhode. It implements multiple functions to determine the value of k. The example below was created by using *dimcalc_share* function with a threshold of 0.5, which sets $k = 2$. As consequence matrix S containing singular values is reduced to two dimensions.

Most students answers in the exam are rated with the maximum points. For this test 20 rated answers are taken, as in the exam most of them achieved the full number of points. The answers are of varying length, the shortest ones contain just five to six words, while the longest consist of two or three sentences with up to thirty or more words. Each of the chosen answers contain a description for all three learning strategies, answers with missing descriptions are ignored.

The evaluation done by the lecturers is used as template to evaluate the results of LSA. It is expected, that these answers have a high similarity to its matching criterion, represented by the documents. The rated answers are interpreted as a query, by using formula (2) the query is projected into the corpus as a pseudo-document and because of their length they be near to the origin of the corpus. To calculate the similarity between the pseudo-documents and the documents, cosine similarities is used.

4 Discussion

Figure 1 (a) shows the corpus with all three assessment criteria (0_Memorization, 1_Organization, 2_Elaboration). It is noticeable that the criterion for memorization lies closer to the origin than the other two criteria. This is a result of the relatively short length of the document which is taken as criterion for memorization. If the similarity between this and the other criteria is calculated, one can see that this is problematic. Document 1_Organization and 2_Elaboration have a cosine similarity of 0.08, so they can be seen as very unequal. While 0_Memorization and 1_Organization have an average similarity of 0.57, criteria 0_Memorization and 2_Elaboration are very similar with a value of 0.87. Therefore and because of the tendency of pseudo-documents to lie close to the origin, it can be expected that using cosine similarity will not be successful. The assessment criterion for the descriptions of the memorization strategy overlaps the criterion for the elaboration strategy.

Looking at precision and recall values proofs this assumption to be correct for the corpus plotted in Figure 1 (a). The evaluation of the answers achieves a recall of 0.62, a precision of 0.51 and an accuracy of 0.68. Although the threshold for a correct rating is set to 0.9, both values can be seen as too low to be used for rating open questions. Since the two criteria for memorization and elaboration

[3] http://cran.r-project.org/web/packages/lsa/index.html

[4] http://tedlab.mit.edu/ dr/SVDLIBC/ This is a reimplementation of SVDPACKC written by Michael Berry, Theresa Do, Gavin O'Brien, Vijay Krishna and Sowmini Varadhan (University of Tennessee).

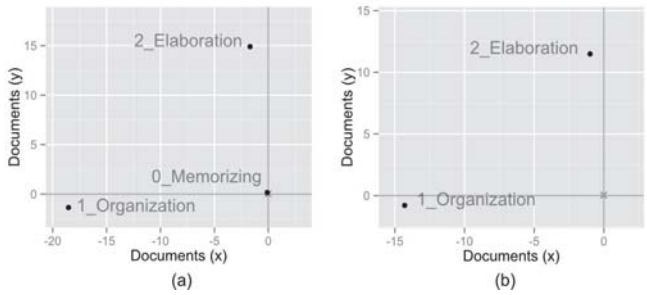

Fig. 1. Figure 1 (a) shows the corpus containing all three assessment criteria. It is illustrated that document 0_Memorizing lies close to the origin. Figure 1 (b) shows the corpus without the document 0_Memorizing. In Figure (a) and (b) the crosses close to the origin mark the positions of the 20 queries.

have a high similarity, a description for one of them gets a high similarity for both criteria. This causes the low precision values for the evaluation.

Figure 1 (b) illustrates the corpus without the document, which is used as criterion for the memorization strategy. Comparing both documents shows a similarity of 0.06. By removing the problematic document from the corpus, the similarity of the students answers to the assessment criterion for elaboration can be calculated without being overlapped by the criterion for memorization. Using this corpus for evaluation improves recall to 0.69, precision to 0.93 and accuracy to 0.83.

If one compares both results, it is remarkable that precision as a qualitative characteristic improves to a high rate, while recall stays at an average level. This means in the context of question rating that answers correctly validated by LSA are very likely rated positive by a human rater. Although LSA creates a precise selection of correct answers, recall rate shows that there are still some positive answers missing in the selection. The increase of accuracy from 0.68 to 0.83 illustrates that the number of true negatives increases by using the second corpus.

5 Conclusion and Future Work

The results of the experiment are encouraging and the general idea of using LSA to rate open questions is functional. The approach of using documents as assessment criterion and project human answers as pseudo-documents into the semantic space constructed by LSA is useful. LSA selects correct answers with a high precision, although some positive rated answers are missing in the selection. But the application shows that some points need to be considered.

All assessment criteria have to be sufficient distinct from each other and should be of a certain length, if cosine similarity is used. As the criterion for

rating the elaboration descriptions shows, it is important that no criterion is overlapped by another. Without considering this, sometimes it is impossible to distinguish which criterion is the correct one. Having a criterion overlapping another one leads to the problem that both criteria get a high similarity, which raises the number of false positives and reduces the precision of the result. This is a mayor difference between the application of LSA as an information retrieval tool or for scoring purposes.

Concerning the average recall value, it is an option to examine the impact of a synonymy dictionary in futher studies. In addition, our result shows that BinIDF weighting works well for a small number of dimensions, as Pincombe (2004) described.

For future work, we plan to use this layout in an online tutorial to perform further tests in winter semester 2013/14. The tutorial is designed as massively multiuser virtual course and will accompany a lecture in educational psychology, which is attended by several hundred students. It will contain two items to gain more empirical evidence and experience with this application and its configuration. To examine the impact on learners long-term memory will be subject to further studies.

References

Deerwester, S., Susan T. D., Furnas, G. W., Landauer, T. K., Harshman, R.: Indexing by latent semantic analysis. Jounal of the American Society For Information Science 41, 391–407 (1990)

Dumais. S. T.: Improving the retrieval of information from external sources. Behavior Research Methods 23, 229–236 (1991)

Fritz, A., Hussy, W., Tobinski, D.: Pädagogische Psychologie. Reinhardt, München (2010)

Jorge-Botana, G., Leon, J. A., Olmos R., Escudero I.: Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora. Journal of Quantitative Linguistics 17, 1–29 (2010)

Landauer, T. K., Dumais, S. T.: Solution to Plato's problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240 (1997)

Landauer, T. K., McNamara, D. S., Dennis, S., Kintsch, W.: Handbook of Latent Semantic Analysis. Routledge, New York and London (2011)

Martin, D. I., Berry, M. W.: Mathematical Foundations Behind Latent Semantic Analysis. Landauer et al., Handbook of Latent Semantic Analysis, 35–55 (2011)

Pincombe, B.: Comparison of human and latent semantic analysis (LSA) judgments of pairwise document similarities for a news corpus (2004)

Higher-order Logic Description of MDPs to Support Meta-cognition in Artificial Agents

Vincenzo Cannella, Antonio Chella, and Roberto Pirrone

Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica,
Viale delle Scienze, Edificio 6, 90100 Palermo, Italy
{vincenzo.cannella26,roberto.pirrone}@unipa.it

Abstract. An artificial agent acting in natural environments needs meta-cognition to reconcile dynamically the goal requirements and its internal conditions, and re-use the same strategy directly when engaged in two instances of the same task and to recognize similar classes of tasks. In this work the authors start from their previous research on meta-cognitive architectures based on Markov Decision Processes (MDPs), and propose a formalism to represent factored MDPs in higher-order logic to achieve the objective stated above. The two main representation of an MDP are the numerical, and the propositional logic one. In this work we propose a mixed representation that combines both numerical and propositional formalism using first-, second- and third-order logic. In this way, the MDP description and the planning processes can be managed in a more abstract manner. The presented formalism allows manipulating structures, which describe entire MDP classes rather than a specific process.

Keywords: Markov Decision Process, ADD, Higer-order logic, u-MDP, meta-cognition

1 Introduction

An artificial agent acting in natural environments has to deal with uncertainty at different levels. In a changing environment meta-cognitive abilities can be useful to recognize also when two tasks are instances of the same problem with different parameters. The work presented in this paper tries to address some of the issues related to such an agent as expressed above. The rationale of the work derives from the previous research of the authors in the field of planning in uncertain environments [4] where the "uncertainty based MDP" (u-MDP) has been proposed. u-MDP extends plain MDP and can deal seamlessly with uncertainty expressed as probability, possibility and fuzzy logic. u-MDPs have been used as the constituents of the meta-cognitive architecture proposed in [3] where the "meta-cognitive u-MDP" perceives the external environment, and also the internal state of the "cognitive u-MDP" that is the actual planner inside the agent. The main drawbacks suffered by MDP models are both memory and computation overhead. For this reason, many efforts have been devoted to define a compact representation for MDPs aimed at reducing the need for computational

resources. The problems mentioned above are due mainly to the need of enumerating the state space repeatedly during the computation. Classical approaches to avoid enumerating the space state are based on either numerical techniques or propositional logic. The first representations for the conditional probability functions and the reward functions in MDPs were numerical, and they were based on decision trees and decision graphs. These approaches have been subsequently substituted by algebraic decision diagrams (ADD) [1][8]. Numerical descriptions are suitable to model mathematically a MDP but they fail to emphasize the underlying structure of the process, and the relations between the involved aspects. Propositional or relational representations of MDPs [6] are variants of the probabilistic STRIPS [5]; they are based on either first-order logic or situation calculus [2] [10][9][7]. In particular, a first-order logic definition of Decision Diagrams has been proposed. In this work we propose a mixed representation that combines both numerical and propositional formalisms to describe ADDs using first-, second- and third-order logic. The presented formalism allows manipulating structures, which describe entire MDP classes rather than a specific process. Besides the representation of a generic ADD as well as the implementation of the main operators as they're defined in the literature, our formalism defines *MetaADDs* (MADD) as suitable ADD abstractions. Moreover, *MetaMetaADDs* (MMADD) have been implemented that are abstractions of MADDs. The classic ADD operators have been abstracted in this respect to deal with both MADDs and MMADDs. Finally, a recursive scheme has been introduced in order to reduce both memory consumption and computational overhead.

2 Algebraic Decision Diagrams

A Binary Decision Diagram (BDD) is a directed acyclic graph intended for representing boolean functions. It represents a compressed decision tree is. Given a variable ordering, any path from the root to a leaf node in the tree can contain a variable just once. An Algebraic Decision Diagram (ADD) [8][1] generalizes BDD for representing real-valued functions $f : \{0, 1\}^n \to \mathbb{R}$ (see figure 1). When used to model MDPs, ADDs describe probability distributions. The literature in this field reports the definition of the most common operators for manipulating ADDs, such as addition, multiplication, and maximization. A homomorphism exists between ADDs and matrices. Sum and multiplication of matrices can be expressed with corresponding operators on ADDS, and suitable binary operators have been defined purposely in the past. ADDs have been very used to represent matrices and functions in MDPs. SPUDD is the most famous example of applying ADDs to MDPs[8].

3 Representing ADDs in Higher-order logic

In our work ADDs have been described in Prolog using first-order logic as a fact in a knowledge base to exploit the Prolog capabilities of managing higher-order logic. An ADD can be regarded as a couple $\langle S, \mathbf{v} \rangle$ where S is the tree's structure,

which is made up by nodes, arcs, and labels, and **v** is the vector containing the values in terminal nodes of the ADD. Let's consider the ADD described in the previous section. Figure 1 shows its decomposition in the structure-vector pair. Terminal nodes are substituted with variables, and the ADD is transformed into its structure. Each element of the vector **v** is a couple made up by a proper variable inserted into the i-th leaf node of the structure, and a probability value that was stored originally into the i-th leaf node.

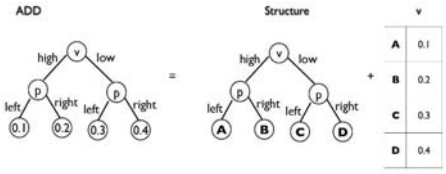

Fig. 1. The decomposition of an ADD in the corresponding structure-vector pair $\langle S, \mathbf{v} \rangle$.

In general, ADDs can be represented compactly through a recursive definition due to the presence of isomorphic sub-graphs. At the same manner, a structure can be defined recursively. The figure 2 shows an example.

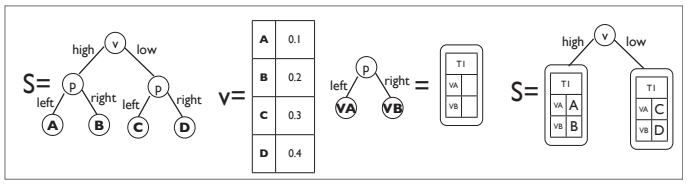

Fig. 2. Each structure can be defined recursively as composed by its substructures. A structure can be decomposed into a collection of substructures. Each substructure can be defined separately, and the original structure can be defined as a combination of substructures. Each (sub)structure is described by the nodes, the labels and the variables in its terminal nodes. Such variables can be unified seamlessly with either another substructure or another variable.

Following this logic, we can introduce the concept of *MetaADD* (MADD), which is a structure-vector pair $\langle S, \mathbf{v} \rangle$ where S is the plain ADD structure, while **v** is an array of variables that are unified with no value (see figure 3). A MADD expresses the class of all the different instances of the same function, which involve the same variables but can produce different results.

An operator op can be applied to MADDs just like in the case of ADDs. In this way, the definition of the operator is implicitly extended. The actual

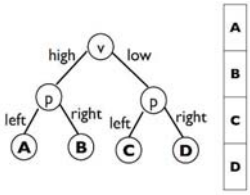

Fig. 3. The MetaADD corresponding to the ADD introduced in the figure 1.

implementation of an operator op applied to MADDs can be derived by the corresponding operator defined for ADDs. Given three ADDs, add_1, add_2, and add_3, and their corresponding MADDs $madd_1$, $madd_2$, and $madd_3$, then $add_1 \,\mathrm{op}\, add_2 = add_3 \Rightarrow madd_1 \,\mathrm{op}\, madd_2 = madd_3$.

The definition of the variables in $madd_3$ depends on the operator. We will start explaining the implementation of a generic operator for ADDs. Assume that the structure-vector pairs for two ADD's are given: $add_1 = \langle S_1, \mathbf{v}_1 \rangle$, $add_2 = \langle S_2, \mathbf{v}_2 \rangle$. Running the operator will give the following result:

$$add_1 \,\mathrm{op}\, add_2 = \langle S_3, \mathbf{v}_3 \rangle$$

Actual execution is split into two phases. At first, the operator is applied to both structures and vectors of the input ADDs separately, then the resulting temporary ADD is simplified.

$$\begin{aligned} S_{temp} &= S_1 \quad \overline{\mathrm{op}} \quad S_2 \\ \mathbf{v}_{temp} &= \mathbf{v}_1 \quad \overline{\mathrm{op}} \quad \mathbf{v}_2 \\ &\mathrm{simplify}(S_{temp}, \mathbf{v}_{temp}) \rightarrow \langle S_3, \mathbf{v}_3 \rangle \end{aligned}$$

Here $\overline{\mathrm{op}}$ is the "expanded" form of the operator where the structure-vector pair is computed plainly. The equations above show that S_{temp} depends only on S_1 and S_2, and it is the same for $\mathbf{v}_{temp}$ with respect to $\mathbf{v}_1$ and $\mathbf{v}_2$.The simplify$(\cdot,\cdot)$ function represents the pruning process, which takes place when all the leaf nodes with the same parent share the same value. In this case, such leaves can be pruned, and their value is assigned to the parent itself. This process is repeated until there are no leaves with the same value and the same parent in any location of the tree (see figure 4). Such a general formulation of the effects produced by an operator on a couple of MADDs can be stored in memory as *Abstract Result* (ABR). ABRs are defined recursively to save both memory and computation too. An ABR is a t-uple $\langle M_1, M_2, Op, M_3, F \rangle$, where M_1, M_2 and M_3 are MADDs, Op is the operator that combines M_1, M_2 and returns M_3, while F is a list of relationships between the variables in M_1, M_2 and M_3, which in turn depend on Op. We applied the abstraction process described so far, to MADDs also by replacing its labels with non unified variables. The resulting structure-vector pairs have been called *MetaMetaADDs* (MMADD) (see figure

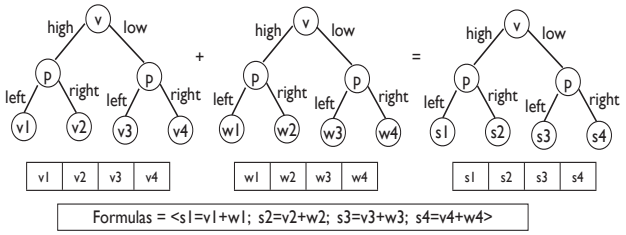

Fig. 4. Two MADDs are added, producing a third MADD. Results are computed according to the formulas described inside the box.

5). We called such computational entity *meta-structure* MS. It has neither values nor labels: all its elements are variables. MS is coupled with a corresponding vector $\mathbf{v}$, which contains variable-label couples (see Figure 5). The definition of

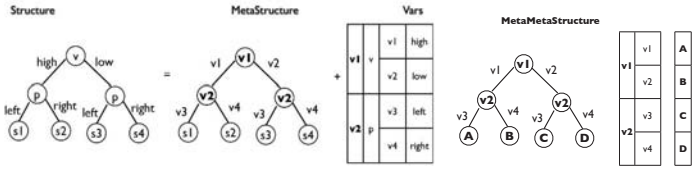

Fig. 5. The decomposition of a MADD structure into the pair $\langle MS, \mathbf{v} \rangle$, and the corresponding MMADD

operators, their abstraction, and the concept of ABR remain unchanged also at this level of abstraction.

4 Discussion of the Presented Formalism and Conclusions

The generalizations of ADDs to second- and third-order logic that were introduced in the previous section, allow managing MDPs in a more efficient way than a plain first-order logic approach. Most part of MDPs used currently, share many regularities in either transition or reward function. As said before, such functions can be described by ADDs. If these regularities appear, ADDs are made up by sub-ADDs sharing the same structures. In these case, computing a plan involves many times the same structure with the same elaboration in different steps of the process. Our formalism allows to compute them only once and save it in a second- and third-order ABR. Every time the agent has to compute structures

that have been used already, it can retrieve the proper ABR to make the whole computation faster. Computation can be reduced also by comparing results in different MDPs. In many cases, two MDPs can share common descriptions of the world, similar actions, or goals, so the results found for a MDP could be suitable for the other one. A second-order description allows comparing MDPs that manage problems defined in similar domains, with the same structure but different values. Finally, a third-order description allows to compare MDPSs, which manage problems defined in different domains but own homomorphic structures. In this case, every second- and third- order ABR computed for the first MDP can be useful to the other one. This knowledge can be shared by different agents. Adding ABRs to the knowledge base enlarges the knowledge of the agent, and reduces the computational effort but implies a memory overhead. Our future investigation will be devoted to devise more efficient ways for storing and retrieving ABRs thus improving the overall performances of the agent.

References

1. R.I. Bahar, E.A. Frohm, C.M. Gaona, G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebraic decision diagrams and their applications. In *Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on*, pages 188–191, 1993.
2. Craig Boutilier, Raymond Reiter, and Bob Price. Symbolic Dynamic Programming for First-Order MDPs. In *IJCAI*, pages 690–700, 2001.
3. Vincenzo Cannella, Antonio Chella, and Roberto Pirrone. A meta-cognitive architecture for planning in uncertain environments. *Biologically Inspired Cognitive Architectures*, 5:1 – 9, 2013.
4. Vincenzo Cannella, Roberto Pirrone, and Antonio Chella. Comprehensive Uncertainty Management in MDPs. In Antonio Chella, Roberto Pirrone, Rosario Sorbello, and Kamilla R. Johannsdottir, editors, *BICA*, volume 196 of *Advances in Intelligent Systems and Computing*, pages 89–94. Springer, 2012.
5. Richard Dearden and Craig Boutilier. Abstraction and approximate decision-theoretic planning. *Artif. Intell.*, 89(1-2):219–283, January 1997.
6. Charles Gretton and Sylvie Thiébaux. Exploiting first-order regression in inductive policy selection. In *Proceedings of the 20th UAI Conference*, UAI '04, pages 217–225, Arlington, Virginia, United States, 2004. AUAI Press.
7. Jan Friso Groote and Olga Tveretina. Binary decision diagrams for first-order predicate logic. *J. Log. Algebr. Program.*, 57(1–2):1 – 22, 2003.
8. Jesse Hoey, Robert St-aubin, Alan Hu, and Craig Boutilier. Spudd: Stochastic planning using decision diagrams. In *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 279–288. Morgan Kaufmann, 1999.
9. Saket Joshi, Kristian Kersting, and Roni Khardon. Generalized first order decision diagrams for first order markov decision processes. In Craig Boutilier, editor, *IJCAI*, pages 1916–1921, 2009.
10. Saket Joshi and Roni Khardon. Stochastic planning with first order decision diagrams. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric A. Hansen, editors, *ICAPS*, pages 156–163. AAAI, 2008.

Dual Aspects of Abduction and Induction

Flavio Zelazek

Department of Philosophy, Sapienza University of Rome, Italy
flavio.zelazek@gmail.com

Abstract. A new characterization of abduction and induction is proposed, which is based on the idea that the various aspects of the two kinds of inference rest on the essential features of increment of (so to speak, extensionalized) comprehension and, respectively, of extension of the terms involved. These two essential features are in a reciprocal relation of duality, whence the highlighting of the dual aspects of abduction and induction. Remarkably, the increment of comprehension and of extension are dual ways to realize, in the limit, a 'deductivization' of abduction and induction in a similar way as the Closed World Assumption does in the case of the latter.

Keywords: abduction, induction, extension, comprehension, duality, closed world assumption.

1 The Uses of Abduction

The analysis of abduction is a topic of very much interest, albeit not a central one, in the fields of philosophy of science and AI. The former often relates abduction to reasoning from effects to cause and to *Inference to the Best Explanation* (IBE), while the latter exploits it in *Abductive Logic Programming* (ALP) [11], closely related to *Inductive Logic Programming* (ILP) [14], which in turn is a powerful tool for Machine Learning.

In the rich subfield of logic programming, the formal characterization of abduction is of particular interest, especially in relation with induction (here the fundamental reference is [5]) and deduction. Typically, abduction is explicated in terms of *default reasoning* and *Negation as Failure* (NAF), which in turn is related to the *completion* technique, which allows talking of abduction in deductive terms.[1]

Now, 20 years after the birth of ALP, there is an impressive proliferation of new frameworks and techniques meant to represent abductive reasoning, so that it seems by now really hard to have precise and yet unifying characterizations of it like those just mentioned. Maybe what is missing is just a simple and general – yet rigorous – enough logical characterization of abduction which encloses the common characters it has across so many diverse approaches and techniques.[2]

[1] For the relevant references see e.g. [7].

[2] In fact the need for unifying frameworks, typical of philosophy and science, is growing also in the fields of computer science and AI: consider, for example, the recent works by Kowalski and Sadri, like [12].

As is well known, the first logical analysis of abduction (and the term itself) has been given in the works of Charles Sanders Peirce. His vast speculation on abduction has been subdivided by Fann [4] into two periods: an early one, from 1859 to 1890, and a later one from 1891 to 1914; to these periods correspond, by and large, the two different conceptions of abduction which have been called by Flach and Kakas [6] the *syllogistic theory* and the *inferential theory*.

What has remained of the first theory, mainly in the AI and logic programming literature, is the triad of examples relating deduction, induction, and abduction (2.623),[3] of which the pertinent one is the following:

$$\frac{\begin{array}{ll}\text{(Rule)} & \text{All the beans from this bag are white}\\ \text{(Result)} & \text{These beans are white}\end{array}}{\text{(Case)} \quad \text{These beans are from this bag}} \qquad (\textsc{Abd}_1)$$

On the other hand, the later peircean theory of abduction is summarized by this notorious reasoning schema (5.189), to which a massive literature is dedicated, especially in the field of philosophy of science:

> The surprising fact, C, is observed;
> But if A were true, C would be a matter of course, $\qquad (\textsc{Abd}_2)$
> Hence, there is reason to suspect that A is true.

The "would be a matter of course" implies something like the fact that A is a *cause* – or at any rate an *explanation* – of C (as this is what generally accounts for C being 'naturally' expected, given A),[4] whence the matching of abduction with (backward) causal reasoning and/or IBE.

Despite the formal clarity of $(\textsc{Abd}_1)$ and the informal intent of $(\textsc{Abd}_2)$, often abduction is represented by this inference schema:

$$\frac{\begin{array}{l}M \to P\\ P\end{array}}{M}\ \textsc{Abd}_0$$

Now, presumably the 'rule' $(\textsc{Abd}_0)$ is simply to be conceived as a forgetful interpretation of $(\textsc{Abd}_2)$, and of this other schema (extracted from $(\textsc{Abd}_1)$), which is at the heart of ALP:

[3] I'm using the standard quotation format for the *Collected Papers* of C. S. Peirce [9]: (<*volume number*>.<*paragraph number(s)*>).

[4] Actually, the quoted expression could also be construed as the weakest claim that C is positively (even if spuriously) correlated to A; but this is usually not the case.

$$\frac{\forall x\ (M(x) \to P(x)) \quad P(s_1)}{M(s_1)}\ \textsc{Abd}_{\textsc{Q}}$$

But the fact is that the schema ($\textsc{Abd}_0$) 'forgets too much': if we want to characterize abduction as IBE, we must operate in a causal/explanatory framework; while if we want to talk of abduction in a purely logical way, e.g. when doing logic programming, we must at least use predicate logic. A formulation in terms of mere propositional logic like that of ($\textsc{Abd}_0$), besides being vaguely evocative, is not sufficient for neither approach; and, what is worst, it leaves the door open to easy criticisms against abduction in general (see note 8 below).

On the other side, it must be said that, often, many useful logical features of reasoning are captured in terms of pure propositional logic; so the problem is: can we characterize abduction within propositional logic in a less trivial and more illuminating way than ($\textsc{Abd}_0$) does? And can we extract from such a characterization any new (or, up to now, largely overlooked) feature of abductive reasoning? The answer to both questions is – I hope – positive, as will be shown in the next section.

2 Abduction as the Dual of Induction

The analysis of abduction – and of its relation of *duality* with induction – which is to follow is based on, and is an explication of, the works belonging to the early logical speculation of Peirce, which are framed in a syllogistic (and, later, probabilistic) setting.

Other works insisting on the duality between abduction and induction are [15], based on the early peircean theory like the present one; [1], which exploits the notion of *preferential entailment* and instead considers the later peircean conception of abduction; [2], which remarks that abduction and induction are related in the same way extension and intension are.[5]

I shall represent induction and abduction by the following inference schemata, which are dual to each other:

[5] Few remarks about the importance of the duality relation. It is a key concept in category theory; in fact, category theory itself seems to have been born to solve a duality problem, with Mac Lane's 1950 article *"Duality in Groups"* (cf. [3]). Moreover, duality is an 'hidden interest' of philosophers: think of Hempel and Oppenheim's attempt to define the *systematic power* ('content') of a theory as the notion which is dual to the one of *logical probability* ('range') in [10]. Finally, the concept of duality is a central one in recent logical research on proof theory and the foundations of theoretical computer science: see [8].

$$\frac{\begin{array}{l} S_1 \vee \cdots \vee S_j \to M \\ S_1 \to P \\ \quad \vdots \\ S_j \to P \end{array}}{M \to P}\ \textsc{Ind}^* \qquad \frac{\begin{array}{l} M \to P_1 \wedge \cdots \wedge P_k \\ S \to P_1 \\ \quad \vdots \\ S \to P_k \end{array}}{S \to M}\ \textsc{Abd}^*$$

These definitions arise as a way to connect the schema ($\textsc{Abd}_1$) above (p. 2), and the analogous schema for induction, to the earliest characterization of inducton and abduction given by Peirce (in (2.511), and also in (2.424–425)), in which *multiple subject/predicate classes* are explicitly inserted.

The natural means (suggested, more or less in clear terms, by Peirce himself in (2.461–516)) to move from a multiplicity of objects to a single class is simply to take their *union* or *intersection*,[6] if they are subjects or, respectively, predicates.

The enlargement of a class by adding members to an union, and the shrinkage of a class by adding members to an intersection, are respectively the extensional counterparts of the concepts of *breadth* and *depth* of a term (i.e. a class), which Peirce investigates thoroughly in (2.407 ff.). They are akin to the concepts of *extension* and *comprehension*,[7] and to the aforementioned notions of *range* and *content* examined in the early hempelian analysis of scientific explanation (see note 5).

The central aspect in the present approach is the idea that induction and abduction are essentially based upon, respectively, the enlargement and the shrinkage of classes, all the other features being derived from these. Console and Saitta [2] make a similar claim, identifying as the logical processes governing induction and abduction those of *generalization* and *specialization*, which are at the base of the theory of Machine Learning.

To see how the inductive and abductive inferences work in this conceptual framework, consider Fig. 1 and Fig. 2, which illustrate what happens in (the set-theoretical versions of) the inferences ($\textsc{Ind}^*$) and ($\textsc{Abd}^*$) with $j = k = 2$. Briefly, an inductive inference is the more confirmed/plausible the more *subjects* it examines, up to the ideal point when the subject class has been enlarged so much that it coincides with the middle term class; at which point all the tests $S_j \to P$ are passed, so that $M \setminus P = \varnothing$ (the shaded area, i.e. the potential

[6] I take for granted the shifting from predicate logic to set notation and then to propositional logic, in the case of categorical propositions of the form "A" (universal affermative); so that $\forall x\ (A(x) \to B(x))$ becomes $A \subseteq B$, which in turn can be written as $A \to B$. Taking $\to, \neg, \vee, \wedge$ instead of $\subseteq, \setminus, \cup, \cap$ (as indeed happens in the schemata ($\textsc{Ind}^*$) and ($\textsc{Abd}^*$)) is really a way to let propositional logic speak for the relevant logical features of objects otherwise represented by predicate logic.

[7] Things are actually more complicated: Peirce makes a distinction between *essential*, *substantial* and *informed* breadth and depth, each subjected to different rules; and he is critical precisely of the careless flattening of these concepts on those of extension and comprehension.

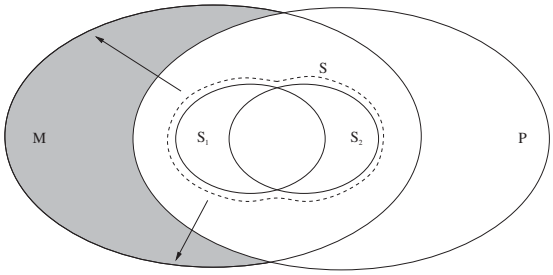

Fig. 1. Induction

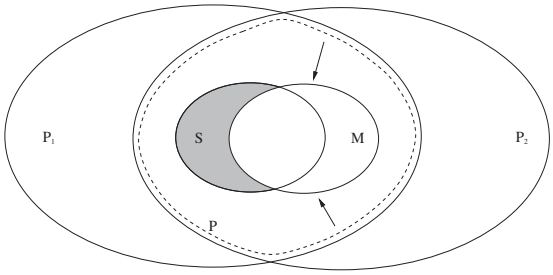

Fig. 2. Abduction

counterexample space for the conclusion, is empty), and the arrow in the first premise of (IND*) can be inverted, so that a deduction is obtained.

Dually, an abductive inference is the more confirmed/plausible the more *properties* of subjects it considers, up to the ideal point when the predicate class has been shrunk so much that it coincides with the middle term class. That being the case, all the tests $S \to P_k$ are passed, so that $S \setminus M = \varnothing$ (again, the shaded area, i.e. the potential counterexample space, is empty), and the arrow in the first premise of (ABD*) can be inverted, so as to get a deduction.

This reversal of arrows in induction and abduction is linked with *completion* or *circumscription* techniques, so basically with the *Closed World Assumption*. On this point an interesting analysis has been made by Lachiche [13].

3 Conclusions and Further Research

The present view of induction and abduction is an 'incremental' one: we increase the number of tests up to the limiting point in which all of the relevant subjects (induction) or all of the relevant properties (abduction) have been tested; even before reaching this ideal point, we get more and more confident about

our conclusion as we proceed further.[8] In the opposite direction, we have as limiting/trivial cases, respectively, single-case induction and (ABD_Q), as in both these inference forms only a single test is taken into consideration.

In this work I have considered implications (i.e. inclusions) as categorical, and tests as yes/no questions; it may be interesting, then, to extend this approach in order to capture probabilistic features as well. This in turn is needed if one wants to define some measure of confirmation/plausibility and, possibly, to implement the new inference figures in a logic programming system.

References

1. Britz, K., Heidema, J., Labuschagne, W.: Entailment, duality, and the forms of reasoning. Tech. Rep. OUCS-2007-01, Department of Computer Science, University of Otago, Otago, New Zealand (2007), available at: `http://www.cs.otago.ac.nz/research/publications/OUCS-2007-01.pdf`
2. Console, L., Saitta, L.: On the relations between abductive and inductive explanation. In: Flach and Kakas [5]
3. Corry, L.: Modern algebra and the rise of mathematical structures. Birkhauser, Basel, 2 edn. (2004)
4. Fann, K.T.: Peirce's theory of abduction. Martin Nijhoff, The Hague (1970)
5. Flach, P.A., Kakas, A.C. (eds.): Abdcution and induction. Essays on their relation and integration, Applied logic series, vol. 18. Kluwer Academic Publishers, Dordrecht (2000)
6. Flach, P.A., Kakas, A.C.: Abductive and inductive reasoning: background and issues. [5]
7. Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.): Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning. Clarendon Press, Oxford (1994)
8. Girard, J.Y.: The Blind Spot. European Mathematical Society, Zürich (2011)
9. Hartshorne, C., Weiss, P. (eds.): Collected papers of Charles Sanders Peirce. The Belknap Press of Harvard University Press, Cambridge, Massachussets (1960)
10. Hempel, C.G., Oppenheim, P.: Studies in the logic of explanation. In: Hempel, C.G.: Aspects of scientific explanation and other essays in the philosophy of science, pp. 245–295. The Free Press, New York (1965)
11. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive Logic Programming. Journal of Logic and Computation 2(6), 719–770 (1992)
12. Kowalski, R.A., Sadri, F.: Towards a logic-based unifying framework for computing (2013), preprint available at: `http://arxiv.org/pdf/1301.6905`
13. Lachiche, N.: Abduction and induction from a non-monotonic reasoning perspective. In: Flach and Kakas [5]
14. Muggleton, S.: Inductive logic programming. Academic Press, London (1992)
15. Wang, P.: From Inheritance Relation to Non-Axiomatic Logic. International Journal of Approximate Reasoning 7, 1–74 (1994)

[8] Thus it can be avoided, or at least minimized, the problem of (ABD_0) committing the *fallacy of affirming the consequent* : an effect can have, e.g., two competing causes/explanations; but it is improbable that for a conjunction of effects (i.e. for a more detailed description of the effect in question – or, in the limit, for its 'maximally detailed', or 'complete', description) both of the same two explanations remain available. I wish to thank a reviewer for having pointed this matter.

Plasticity and Robotics

Martin Flament Fultot

SND, Université Paris-Sorbonne / CNRS, Paris
martin.flament-fultot@paris-sorbonne.fr

Abstract. The link between robotic systems and living systems is increasingly considered to be important. However finding out more precisely which properties these two kinds of system must share is a difficult question to answer. It is suggested that behavioral plasticity constitutes a crucial property that robots must share with living beings. A classification is then proposed for the different aspects of plasticity that contribute to global behavioral plasticity in robotic and living systems. These are mainly divided into four dimensions and three orders. Finally some consequences of this classification are mentioned regarding the future of biologically-inspired robotics and the role of evolutionary AI.

Keywords: Plasticity, Adaptation, Biologically-Inspired Robotics, Artificial Intelligence, Artificial Life, Cognitive Science, Philosophy

1 Introduction

More than ten years ago, Rodney Brooks observed the state of AI and A-life and concluded that, although important progress had been made, something fundamental was still missing from robotics [1]. The symptom: robots don't quite look like living things yet. Paradoxically, after more than a decade of intense research, there has been very interesting and promising developments–some robots now do look more like living creatures–yet we are still missing the special ingredient, i.e. an explicit general principle responsible for this. Moreover, some of the best looking (i.e. more life-like looking) robots can behave and look very differently, making it very unlikely to associate the general principle to any particular design. So what can this general principle be–if there is any single principle at work ? Is it better cognition: a richer representation of the world, more computing power, more efficient algorithms ? Or is it better bodies: more realistic morphologies, lighter materials, more powerful actuators ? I propose that the principle we are looking for is *plasticity*.

Robots that remind us of the living do so because they are plastic. Notice that this a property of their *behavior*–whenever structure also reminds of the living it is only insofar as it contributes to life-like behavior. So it is neither structure nor function that renders some robots plastic but rather *how* they carry out their function and how individual factors contribute to overall plasticity. Before going any further an operational definition of plasticity has to be provided. Thus I take plasticity to be *the potentially adaptive capacity to change one's behavior*

at some level. In other words the capacity to modify one's state under some aspect towards the accomplishment of a given task in a given task-environment. In the rest of this paper I will propose a general classification of the different kinds of plasticity an agent can be endowed with without pointing at any specific mechanism since these kinds of plasticity seem to be multiply realizable. This classification is only intended as a provisional frame to start tackling a concept that hasn't been explicitly and systematically studied until now. Any changes and improvements that might come up in the future are highly welcome.

2 Plasticity and Plasticities

I divide plasticity into *Dimensions*, *Orders* and *Levels*. On the Dimension axis we find the traditional factors of interaction: environment, body and cognition. These are mainly inspired by the idea stemming from embodied AI that behavior emerges from the interactions between these factors [2],[3]. I add a fourth dimension, viz. development, understood here as the process of building the agent, putting together its constituent elements, usually through endogenous growth. Development is increasingly considered to be an essential factor for the design of robotic agents and there has been different attempts to integrate it with the principles of AI, although it is still a complicated project to accomplish [4]. Some consider development as a temporal dimension shared with learning. My treatment will be different as learning consists in an Order rather than a Dimension.

There are at least 3 orders of plasticity. Peter Godfrey-Smith [5] introduces the first two orders to describe the complexity of a given organism. First order plasticity thus refers to the capacity to produce different behaviors according to the situation, e.g. reactive behavior, such as escaping a predator when detected. Second order plasticity refers to the capacity to change the rules linking those behaviors to those situations, e.g. learning as in adding a new rule to escape some animal if it did some harm. To these two orders of plasticity I add a third one–0.5 order plasticity–between the first and pure lack of reaction since it is sometimes adaptive to not resist yet not react actively either. I call it 0.5–even though it doesn't mean much mathematically–because it is a *quasi* first order plasticity. The idea will become clearer below. Next I will provide some mechanisms belonging to the living and artificial world as illustrations of the principles of plasticity and the classification I propose. Then I will conclude with a few remarks about the future of robotics.

3 Plasticity: A Classification

3.1 Cognitive Plasticity

This is the most accepted and intuitive dimension where plasticity is found. Consider as an example of first order cognitive plasticity the main computational element: programs. Programs hold instructions or action rules, mapping inputs

to outputs. Thus a given agent controlled by a program can show first order plasticity by responding with different actions (outputs) to different situations (inputs). Second order cognitive plasticity, on the other hand, is simply the capacity to learn. Our first order program, for instance, could hold instructions to evaluate the success of some of its mappings from input to output and change them to improve performance.

0.5 order cognitive plasticity is the adaptive modification of the agent's cognitive apparatus by some *direct* force or effect from the environment, without mediating internal states. Consider, for instance, Bird and Layzell's evolved radio [6]. This was an evolved electronic circuit which had been repeatedly selected for its capacity to produce stable oscillatory outputs. But the authors soon discovered that the oscillations actually came from direct electromagnetic induction from a nearby computer. So-called oscillatory entrainment also happens in neural networks where the neural units are highly interconnected and form reverberating circuits [7]. Insofar as the oscillating frequency is passively adopted from an external source, this is 0.5 order plasticity. Other well-known examples can be found in the direct chemical action of neuromodulators added to the system, such as drugs and other substances (see [8] on the behavior of GasNets, for instance).

3.2 Bodily Plasticity

Embodied cognition has contributed profoundly to placing the body back in the main field of behavioral causation [9], [18]. The body too presents all three orders of plasticity which must be taken under consideration when trying to understand cognitive phenomena and particularly when designing AI agents. 0.5 order bodily plasticity could be divided into materials and morphology. Materials can thus be soft and *comply* passively to external forces as in robotic arms which yield to forces thanks to rubber components [10]. Another way to increase this kind of plasticity is to have many degrees of freedom and actuators which increase the number of different morphological configurations and movements the agent can afford.

First order bodily plasticity is less conspicuous. Still it can be found in the structural and material properties of the body which don't just passively yield to external forces, but actively react, adding some mediating process or mechanical action. Muscles and tendons, for instance, are increasingly being added to legged robots since they can store energy and go back by themselves to their preferred configuration like springs [11]. The body anatomy's intrinsic dynamics too can contribute to plastic behavior. Radical demonstrations of this capacity include the famous Dynamic Passive Walker, which literally walked through an inclined plane without any actuators nor control system [12], the passive somersault agent [13], and more recently IHMC's Fast Runner leg design which mechanically handles most of the robot's gait. Biomechanical limb coordination can provide mediating states typical of first order plasticity.

Finally a growing literature on muscle memory shows how muscle performance can change over time in order to adapt to circumstances, thus allowing for

second order bodily plasticity. For instance, muscles having a well-differentiated function can change drastically if innervated differently, cumulative effects due to an increasing demand in energy spending results in a modification of the enzymatic activity of muscular cell's mitochondria, muscle capillary density varies according to exercise in order to adjust oxygen levels, and muscle's architecture changes following repeated use [14]. All these effects are characteristic of second order plasticity since they are state fluctuations happening over iterations of behavioral episodes. Nothing of the sort is, to my knowledge, applied in robotics as of now.

3.3 Developmental Plasticity

I propose to separate this dimension from the cognitive and the bodily dimensions since 1) cognitive processes are reversible while developmental processes are not [15]; 2) genetic factors don't play a major role during cognitive tasks while they tend to be central during development; and 3) cognitive and bodily factors use material already present to the system while development consists mainly in the fabrication or modification of new tissue.

When talking about first order developmental plasticity the best examples are norms of reaction and polyphenism which are now well-known and increasingly studied phenomena [16]. The tadpoles in some species of frog can detect predatory presence in their environment and develop particular tissues oriented to protection (Ref. [16] p. 209). Godfrey-Smith [5] holds that this kind of plasticity shows proto-cognitive properties since the system does not simply passively yield to, say, the predator's presence, but instead detects it and executes some adapted developmental routine.

In order to distinguish first order from 0.5 order developmental plasticity one must keep in mind that development is a chemical process. As such there can be many sources of direct action over chemical conditions leading to variability in the final phenotype. Some species of fly's growth speed, for instance, depends on temperature. Insofar as no genetic factors are involved in detecting the temperature and triggering some specific reaction, this form of plasticity can be due to direct catalytic action from the heat [17].

Concerning second order developmental plasticity there is a fundamental difference between the developmental dimension and the other two. Indeed, second order variations don't happen during the lifetime of the agent. This can be a consequence of the already mentioned irreversibility of this dimension. Second order variations then seem to concern the genome when it is replicated. Crossing over, for instance, can be seen as a mechanism whose function is to increase second order developmental plasticity. Nevertheless, robots' life cycles are not necessarily restricted to be equal to those of known biological agents. Second order plasticity during the lifetime of a robot is not conceptually impossible.

3.4 Environmental Plasticity

As stated earlier, environmental plasticity intervenes during behavior. A case of 0.5 order environmental plasticity can be found in the so-called Swiss Robots from Rolf Pfeifer's lab, which have an architecture similar to Braitenberg's vehicles'. The difference is that in the case of the Swiss Robots, the environment *contributes* to the behavior and the task. Indeed, by being passively moved by the robots, blocks change the architecture of the environment, thus affecting the behavior of the robots, and producing the emergent result of a well ordered environment where the blocks are clustered together instead of randomly distributed over the arena [18].

First order environmental plasticity can easily be produced by adding other agents to the environment–plasticity will be inherited from the living plastic elements present in the world. In addition there are other objects such as scaffolds and other kinds of inanimate elements in a given environment that can count as first order environmental plasticity. For instance some monkeys use the spring-like properties of tree branches in order to jump from tree to tree. Also many complex inanimate phenomena are not just passive reactions to, e.g., a strong sound or perturbation as in avalanches, but rather relatively long processes that could be used adaptively by an agent in some plausible scenarios such as attempting to bury an enemy under the snow.

Finally second order environmental plasticity can be defined as *any cumulative effect in the environment leading to the progressive modification of an agent's behavior over consecutive episodes*. Stigmergies constitutes a proverbial case here. These are commonly known as the trails of pheromones ants leave behind when navigating an environment and which, upon attracting other ants to follow the scent, progressively increase its concentration levels thus creating a road which affects the overall behavior of ants over time. Stigmergies can also be obtained with mere inorganic soil if it can cumulate depth when walked upon.

4 Consequences and Conclusion

How can all these forms of plasticity be integrated in a single agent-environment system ? There seems to be just too many interactions to track in a functioning robotic agent. But this is precisely the answer to Brooks' question about the extra ingredient: life-like behavior is a property of multiply plastic integrated agents, such as animals. So plasticity alone doesn't guarantee adaptiveness. The classification shows that many aspects of plasticity need to be carefully tuned and integrated in order to obtain a functional agent. One way to ensure that plasticity contributes to adaptiveness is to seek some sort of balance between the dimensions and orders of plasticity [18]. Obtaining such a balance from *a priori* design is extremely difficult. Nevertheless submitting the agent to a selective pressure should guarantee a balanced result, as it is the case with living creatures. Some promising work is already being carried out in this direction (e.g. [19]). This implies that evolutionary robotics is destined to fulfill a crucial role in AI, by enhancing and increasing our techniques and knowledge about evolutionary and

learning algorithms directed towards the production of plastic, life-like agents. Additional progress must be expected from new materials and computational power to simulate realistic agent-environment systems when development is too expensive to reproduce in real robots.

References

1. Brooks, R.A.: The relationship between matter and life. Nature. 409, 409-411 (2001)
2. Brooks, R.A.: Intelligence Without Representation. Artificial Intelligence. 47, 139-159 (1991)
3. Nehmzow, U.: Scientific Methods in Mobile Robotics: Quantitative Analysis of Agent Behaviour. Springer (2006)
4. Lungarella, M., Metta, G., Pfeifer, R., Sandini, G.: Developmental robotics: a survey. Connection Science. 15, 151–190 (2003)
5. Godfrey-Smith, P.: Complexity and the Function of Mind in Nature. Cambridge University Press (1996)
6. Bird, J., Layzell, P.: The evolved radio and its implications for modelling the evolution of novel sensors. Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC 02. pp. 1836–1841 (2002)
7. Buzsáki, G.: Rhythms of the Brain. Oxford University Press, USA (2006)
8. Husbands, P., Philippides, A., Smith, T., OShea, M.: The Shifting Network: Volume Signalling in Real and Robot Nervous Systems. In: Kelemen, J. and Sosk, P. (eds.) Advances in Artificial Life. pp. 23-36. Springer Berlin Heidelberg (2001)
9. Clark, A.: Being There: Putting Brain, Body, and World Together Again. MIT Press (1997)
10. Michie, D., Johnson, R.: The Creative Computer. Penguin Books (1985)
11. Pfeifer, R.: On the Role of Morphology and Materials in Adaptive Behavior. Presented at the SAB-6, Proc. of the 6th Int. Conf. on Simulation of Adaptive Behavior (2000)
12. McGeer, T.: Passive Dynamic Walking. The International Journal of Robotics Research. 9, 62–82 (1990)
13. Raibert, M., Playter, R., Ringrose, R., Bailey, D., Leeser, K.: Dynamic legged locomotion in robots and animals. NASA STI/Recon Technical Report N. 96, 17141 (1995)
14. Bottinelli, R., Reggiani, C.: Skeletal Muscle Plasticity in Health and Disease: From Genes to Whole Muscle. Springer (2006)
15. Piersma, T., Drent, J.: Phenotypic flexibility and the evolution of organismal design. Trends in Ecology Evolution. 18, 228–233 (2003)
16. Gilbert, S.F.: The Genome in Its Ecological Context. Annals of the New York Academy of Sciences. 981, 202–218 (2002)
17. Gotthard, K., Nylin, S., Nylin, S.: Adaptive Plasticity and Plasticity as an Adaptation: A Selective Review of Plasticity in Animal Morphology and Life History. Oikos. 74, 3 (1995)
18. Pfeifer, R., Bongard, J.: How the Body Shapes the Way We Think: A New View of Intelligence. MIT Press (2007)
19. Bongard, J.: Morphological change in machines accelerates the evolution of robust behavior. Proceedings of the National Academy of Sciences. 108, 1234–1239 (2011)

Characterising citations in scholarly articles: an experiment

Paolo Ciancarini[1,2], Angelo Di Iorio[1], Andrea Giovanni Nuzzolese[1,2], Silvio Peroni[1,2], and Fabio Vitali[1]

[1] Department of Computer Science and Engineering, University of Bologna (Italy)
[2] STLab-ISTC, Consiglio Nazionale delle Ricerche (Italy)
ciancarini@cs.unibo.it, diiorio@cs.unibo.it, nuzzoles@cs.unibo.it, essepuntato@cs.unibo.it, fabio@cs.unibo.it

Abstract. This work presents some experiments in letting humans annotate citations according to CiTO, an OWL ontology for describing the function of citations. We introduce a comparison of the performance of different users, and show strengths and difficulties that emerged when using that particular model to characterise citations of scholarly articles.

Keywords: CiTO, act of citing, citation function, ontology, scholarly articles, semantic publishing, user testing

1 Introduction

The mere existence of a citation might not be enough to capture the relevance of the cited work. For instance, some simple questions arise: is it correct to count negative and positive citations in the same way? Is it correct to give self-citations the same weight of others? Is it correct to give a survey the same weight of a seminal paper, by only counting the number of times it has been cited? A more effective characterisation of citations opens interesting perspectives that go beyond the quantitative evaluation of research products, as highlighted in [2].

To this end, the first issue to address is to identify a formal model for characterising the nature of citations in a precise way – i.e. a citation model. The citation model has to capture the citation functions, i.e. "the author's reasons for citing a given paper" [8]. Even assuming that such a citation model exists and is well established, the task of annotating citations with their citation functions is very difficult from a cognitive point of view. First, the "citation function is hard to annotate because it in principle requires interpretation of author intentions (what could the author's intention have been in choosing a certain citation?)" [7]. Second, one has to create his/her own mental model of the citation model, so as to associate a particular meaning to each of the various functions defined by the citation model. Third, one has to map, by means of the mental model, the personal interpretation of author's intention emerging from a written text containing a citation with the one of the functions of the citation model.

Our work is positioned within the field of "Semantic Web and Cognition". In particular, the goal of this paper is to analyse weaknesses and strengths of

a particular citation model, studying how it has been used (and misused) by the users for the annotation of citations. The model under investigation is CiTO (Citation Typing Ontology)[3] [5], an OWL ontology for describing the nature of citations in scientific research articles and other scholarly works. We present the results of a preliminary user testing session with five users to whom we asked to assign CiTO properties to the citations in the Proceedings of Balisage 2011.

The paper is then structured as follows. In Section 2 we introduce previous works on classification of citations. In Section 3 we present our experimental setting and results: we go into details of the analysis performed by the humans and discuss the outcomes. Finally we conclude the paper sketching out some future works in Section 4.

2 Related works

Teufel et al. [7] [8] study the function of citations – that they define as "author's reason for citing a given paper" – and provide a categorisation of possible citation functions organised in twelve classes, in turn clustered in Negative, Neutral and Positive rhetorical functions. Jorg [3] analysed the ACL Anthology Networks[4] and found one hundred fifty cue verbs, i.e. verbs usually used to carry important information about the nature of citations: based on, outperform, focus on, extend, etc. She maps cue verbs to classes of citation functions according to the classification provided by Moravcsik et al. [4] and makes the bases to the development of a formal citation ontology.

These works actually represent some of the sources of inspiration of CiTO (the Citation Typing Ontology) developed by Peroni et al. [5], which is the ontology we used in our experiment. CiTO permits the motivations of an author when referring to another document to be captured and described by using Semantic Web technologies and languages such as RDF and OWL.

3 Using CiTO to characterise citations

In order to assess how CiTO is used to annotate scholarly articles, we compared the classifications performed by humans on a set of citations. The role of CiTO in such a process was obviously prominent. We in fact used the experiment to study the effectiveness of CiTO, to measure the understandability of its entities, and to identify some possible improvements, extensions and simplifications.

Our goal was to answer to the following four research questions (RQs):

1. How many CiTO properties have been used by users during the test?
2. What are the most used CiTO properties?
3. What is the global inter-rater agreement among users?
4. What are the CiTO properties showing an acceptable positive agreement between users?

[3] CiTO: http://purl.org/spar/cito.
[4] ACL Anthology Network: http://clair.eecs.umich.edu/aan/index.php.

The test bed includes some scientific papers encoded in XML DocBook, containing citations of different types. The papers are all written in English and chosen among those published in the proceedings of the Balisage Conference Series (devoted to XML and other kinds of markup). We automatically extracted citation sentences, through an XSLT transform, from all the papers published in the seventh volume of the proceedings, which are freely available online[5]. The XSLT transform is available at http://www.essepuntato.it/2013/citalo/xslt.

We took into account only those papers for which the XSLT transform retrieved at least one citation (i.e. 18 papers written by different authors). The total number of citations retrieved was 377, for a mean of 20.94 citations per paper. We then filtered all the citation sentences that contain verbs (extends, discusses, etc.) and/or other grammatical structures (uses method in, uses data from, etc.) that carry explicitly a particular citation function. We considered that rule as a strict guideline as also suggested by Teufel et al. [7]. We obtained 104 citations out of 377, obtaining at least one citation for each of the 18 paper used (with a mean of 5.77 citations per paper). These citations are very heterogeneous and provide us a significative sample for analysing human classifications. Finally, we manually expanded each citation sentence (i.e. the sentence containing the reference to a bibliographic entity) selecting a context window[6], that we think is useful to classify that citation.

3.1 Results

The test was carried on, through a web interface, by five users, all academic but not necessarily expert in Computer Science (the main area of the Balisage Conference). None of them was an expert user of CiTO. Each user processed each citation sentence separately, with its full context window, and had to select one CiTO property for that sentence. Users could also revise their choices and perform the experiments off-line. There was no time constraint and users could freely access the CiTO documentation. We used R[7] to load the data and elaborate the results. All the data collected are available online at http://www.essepuntato.it/2013/aic2013/test.

The experiments confirmed some of our hypotheses and highlighted some unexpected issues too. The first point to notice is that our users have selected 34 different CiTO properties over 40, with an average of 22.4 properties per user (RQ1). Moreover a few of these properties have been used many times, while most of them have been selected in a small number of cases, as shown in Table 1 (RQ2). There were 6 properties not selected by any user: compiles, disputes, parodies, plagiarizes, refutes, and repliesTo.

These data show that there is a great variability in the choices of humans. In fact only 3 citations (out of 104) have been classified with exactly the same

[5] Proceedings of Balisage 2011: http://balisage.net/Proceedings/vol7/cover.html.

[6] The context window [6] of a citation is a chain of sentences implicitly referring to the citation itself, which usually starts from the citation sentence and involves few more subsequent sentences where that citation is still implicit [1].

[7] R project for statistical computing: http://www.r-project.org/.

Table 1. The distribution of CiTO properties selected by the users.

# Citations	CITO property
110	citesForInformation
39	citesAsRelated
38	citesAsDataSource
32	citesAsAuthority, obtainsBackgroundFrom
28	citesAsEvidence, citesAsSourceDocument
24	obtainsSupportFrom
23	citesAsRecommendedReading, usesMethodIn
21	citesAsPotentialSolution
< 21	agreesWith, citesAsMetadataDocument, containsAssertionFrom, credits, critiques, discusses, documents, extends, includesQuotationFrom, usesConclusionsFrom
< 5	confirms, corrects, derides, disagreesWith, includesExcerptFrom, qualifies, retracts, reviews, ridicules, speculatesOn, supports, updates, usesDataFrom

CiTO property by all 5 users, while for 23 citations the humans selected at most two properties. These results are summarised in Table 2, together with the list of selected properties. In that table, we indicate how many citations of the dataset users agreed, and the number of properties selected by the users.

Table 2. The distribution of citations and CiTO properties on which users agreed.

Max # of properties per citation	# Citations in the dataset	CiTO properties
1	3	citesAsDataSource (5), citesAsPotentialSolution (5), citesAsRecommendedReading (5)
2	23	citesForInformation (27), citesAsDataSource (21), citesAsRelated (16), citesAsRecommendedReading (11), citesAsPotentialSolution (9), citesAsAuthority (6), credits (4), includesQuotationFrom (4), critiques (3), discusses (3), obtainsBackgroundFrom (3), usesMethodIn (3), citesAsSourceDocument (2), obtainsSupportFrom (2), citesAsEvidence (1)

3.2 Evaluation

Considering all the 104 citations, the agreement among humans was very poor. We measured the Fleiss' k (that assesses the reliability of agreement between a fixed number of raters classifying items) for the 5 raters over all 104 subjects and obtained k = 0.16, meaning that there exists a positive agreement between users but it is very low (RQ3). However there exists a core set of CiTO properties whose meaning is clearer for the users and on which they tend to agree. In fact, even considering the whole dataset whose k value was very low, we found a moderate positive local agreement (i.e. 0.33 <= k <= 0.66) on some proper-

ties (RQ4): citesAsDataSource (k = 0.5), citesAsPotentialSolution (k = 0.45), citesAsRecommendedReading (k = 0.34), includesQuotationFrom (k = 0.49).

The results on the core CiTO properties were also confirmed by a slightly different analysis. We filtered only the 23 citations on which the users used at most two properties, as mentioned earlier in table Table 2. The k value on that subset of citations showed a moderate positive agreement between humans (k = 0.55, with 5 raters over 23 subjects). We had also moderate and high local positive agreement (i.e. k > 0.66) for 10 of the 15 properties used. The 5 properties showing an high positive agreement are citesAsDataSource (k = 0.77), citesAsPotentialSolution (k = 0.88), citesAsRecommendedReading (k = 0.7), credits (k = 0.74, that was not included in the core set mentioned above), and includesQuotationFrom (k = 0.74); the properties showing a moderate positive agreement are citesAsRelated (k = 0.6), citesForInformation (k = 0.4), critiques (k = 0.49), obtainsBackgroundFrom (k = 0.49), and usesMethodIn (k = 0.49).

3.3 Discussion

One of our findings was that some of the properties were used only few times or not used at all. This result can depend on a variety of factors. First, the authors of the articles in our dataset, which are researchers on markup languages, use a quite specific jargon so the citation windows resulted not easy to interpret with respect to citations. Second, the positive or negative connotation of the properties was difficult to appreciate. For instance, the fact that the properties carrying negative judgements (corrects, derides, disagreesWith, etc.) are less frequent than the others supports the findings of Teufel et al. [7] on this topic.

Although we think the intended audience of the research articles one chooses for such an experiment may bias the use of some properties, we also believe that some properties are actually shared among different scholarly domains. The property citesForInformation is a clear example. As expected, it was the most used property, being it the most neutral of CiTO. This is in line with the findings of Teufel et al. [8] on the analysis of citations within Linguistics scholarly literature, where the neutral category Neut was used for the majority of annotations by humans. Although its large adoption, citesForInformation had a very low positive local agreement (k = 0.13). This is not surprising since the property was used many times, often as neutral classification on citations that were classified in a more precise way by other users.

One of the reasons for having a low positive agreement in total (i.e. k = 0.16) could be the high number of properties (40) defined in CiTO. To test this, we mapped the 40 CiTO properties into 9 of the 12 categories identified by Teufel et al. [8][8] and re-calculated the Fleiss' k obtaining k = 0.19. Even if the agreement is slightly better than the one we got initially, the number of available choices did not impact too much. It seems to be only one of the factors to take into account for that low agreement. Another important factor might have been the

[8] The alignments of the forty CiTO properties with Teufel et al.'s classification is available at http://www.essepuntato.it/2013/07/teufel.

flat organisation of CiTO properties. Since there is no hierarchical structure, each user followed its own mental mapping and ended up selecting very different values – probably because users' mental models differed largely between users.

We also asked humans informally what were the cognitive issues they experienced during the test. Some of them highlighted that it was easy to get lost in choosing the right property for a citation because of the large number of possible choices. In addition, they also claimed that supporting the documentation of CiTO with at least one canonical example of citation for each property could be useful to simplify the choice.

4 Conclusions

The main conclusion for this paper is that classifying citations is an extremely difficult job also for humans, as demonstrated in our experiments on the properties of CiTO. The human analysis we presented herein gave us important hints on the understanding and adoption of CiTO, still showing some uncertainty and great variability. The identified strengths and weaknesses will be used to further improve the ontology, together with experiments on a larger set of users, decreasing the number of possible choices (for instance by using only the CiTO properties showing more agreement among humans).

References

1. Athar, A., Teufel, S. (2012). Detection of implicit citations for sentiment detection. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: 18-26.
2. Ciancarini, P., Di Iorio, A., Nuzzolese, A. G., Peroni, S., & Vitali, F. (2013). Semantic Annotation of Scholarly Documents and Citations. To appear in Proceedings of 13th Conference of the Italian Association for Artificial Intelligence (AI*IA 2013).
3. Jorg, B. (2008). Towards the Nature of Citations. In Poster Proceedings of the 5th International Conference on Formal Ontology in Information Systems.
4. Moravcsik, M. J., Murugesan, P. (1975). Some Results on the Function and Quality of Citations. In Social Studies of Science, 5 (1): 86-92.
5. Peroni, S., Shotton, D. (2012). FaBiO and CiTO: ontologies for describing bibliographic resources and citations. In Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 17 (December 2012): 33-43. DOI: 10.1016/j.websem.2012.08.001
6. Qazvinian, V., Radev, D. R. (2010). Identifying non-explicit citing sentences for citation-based summarization. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: 555-564.
7. Teufel, S., Siddharthan, A., Tidhar, D. (2006). Automatic classification of citation function. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing: 103-110.
8. Teufel, S., Siddharthan, A., Tidhar, D. (2009). An annotation scheme for citation function. In Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue: 80-87.

A meta-theory for knowledge representation

Janos J. Sarbo

Radboud University, ICIS
Nijmegen, The Netherlands

Abstract. An unsolved problem in AI is a representation of meaningful interpretation. In this paper we suggest that a process model of cognitive activity can be derived from a Peircean theory of categories. By virtue of the fundamental nature of categories, the obtained model may function as a meta-theory for knowledge representation.

1 Introduction

An unsolved problem in AI is a representation of meaningful interpretation. The complex nature of this problem is illustrated by Searle's famous Chinese room argument thought experiment (CRA) [6]. Throughout the CRA debate Searle maintained that meaningful (semantic) and computational (syntactic) interpretation must be qualitatively different.

From the perspective of knowledge representation (KR) we may identify two extreme positions in the reaction by computer science on the above problem of AI. According to the first one, meaningful are those concepts that have that property by definition. Traditional theories of KR, in a broad sense, including program specification and theorem proving, facilitate this conception. In our view, the underlying reasoning may not be correct. Although individual concepts obtained by human activity can be meaningful, a combination of such concepts may not possess that property. This is a consequence of the inadequacy of the used ontology for a definition of genuine meaningfulness (we will return to this point in the next section) and the possibility of a combination of concepts of arbitrary length in KR (in the lack of a definition we may not be able to derive if a combination of concepts is meaningful). According to the second position above, meaningful concepts arise through interpretation (hence meaningful interpretation is a tautology). Following this conception, a representation of (meaningful) interpretation is in need of a paradigmatically new ontology, enabling meaningful and not-meaningful to be represented qualitatively differently.

In this paper we elaborate on the second position above, and how this view can be supported computationally. To this end we consider the question what is involved in meaningful interpretation. For, even if we may not be able to capture the real nature of interpretation, knowledge about its properties may allow us to build computer programs approximating and thereby enhancing human processing, e.g., through simulating the operations involved in it.

Below we begin with an analysis of traditional KR. We return to an overview of a novel ontology and knowledge representation, in Sect. 3.

2 Traditional knowledge representation

As meaningful interpretation is our common experience, its properties must be respected by models of genuine human processing. In this section we suggest that traditional KR may not be able to comply with this requirement and that, some of the problems in computer science could be a consequence of the above deficiency of traditional modeling as well. A property shared by traditional theories of KR is their foundation in the Aristotelian categorical framework. Aristotle's ten categories can be distinguished in two qualitatively different types: unique *substances*, that are independent; and accidental categories or *attributes*, such as quantity, quality and relation, that are 'carried' by a substance. Clearly, in the Aristotelian framework, actual and meaningful attributes (hence also such substance–attribute relations) cannot be represented in a qualitatively different fashion. From this we conclude that his ontology may not be satisfactory for the definition of a model of authentic interpretation.

Notably the same problem, the lack of a suitable ontology, seems to have been the driving force behind important discoveries in knowledge modeling, in the past. An example is the problem of program specification, revealed by E.W. Dijkstra, in 1968. By virtue of the possibility of an unbridled use of 'goto' statements, enabled by programming languages at that time, programs were frequently error-prone. Dijkstra suggested a systematic use of types of program constructs, which he called Structured Programming. Briefly, this states that three ways of combining programs –sequencing, selection, and iteration (or recursion)– are sufficient to express any computable function. Another example is the problem of an apparent diversity of models of natural language syntax, exposed by A.N. Chomsky, in 1970. In his X-bar theory, Chomsky claimed that among their phrasal categories, all human languages share certain structural similarities, that are lexical category, relation, and phrase.

In our view, the trichotomic character of classification, illustrated by the examples above, may not be accidental. We foster the idea that a representation of meaningful concepts, and in general, the definition of a model of meaningful interpretation asks for a three-categorical ontology. A theory satisfying the above condition can be found in the categorical framework by C.S. Peirce (1839-1914). By virtue of the fundamental nature of categories, and the relation between Peirce's categories and his signs, Peircean theory is considered by many to be a theory of the knowable hence a meta-theory for knowledge representation.

3 Towards a new ontology

According to Peirce [3], phenomena can be classified in three categories, that he called firstness, secondness, and thirdness. Firstness category phenomena involve a monadic relation, such as the relation of a quality to itself. Secondness category phenomena involve a dyadic relation, such as an actual (or ad-hoc) relation between qualities. Thirdness category phenomena involve a triadic relation, such as an interpretation of a relation, rendering an explanation or a reason to it,

thereby generating a meaningful new concept. The three Peircean categories are irreducible, for example, triadic relations cannot be decomposed into secondness category actual relations. From a KR perspective, the categories can be considered to be qualitatively different. For instance, secondness is qualitatively less meaningful than thirdness. Conform its relational character, triadic classification can be applied recursively. Below, a category can be designated by its ordinal number, e.g., secondness by the integer '2'.

Our examples, in Sect. 2, exhibit the aspects of Peirce's three categories. A sequence, a lexical item, are independent phenomena, exhibiting the aspect of firstness (1). A selection between alternatives, that are involved, a language relation, defined by constituent language symbols, e.g., in a syntactic modification structure, are relation phenomena, exhibiting the aspects of secondness (2). An iteration, abstracting alternatives and sequences of instructions into a single instruction, a phrase, merging constituent expressions into a single symbol, are closure phenomena, exhibiting the aspects of thirdness (3).

Peirce's three categories are related to each other according to a relation of dependency: categories of a higher ordinal number involve a lower order category. A distinguishing property of the Peircean categorical schema is that only thirdness can be experienced, firstness may only appear through secondness, and secondness only through thirdness. This subservience relation of the three categories implies that categories of a lower ordinal number evolve to hence need a higher order category.

The sample classifications, in Sect. 2, satisfy the conditions of dependency between the categories. For instance, an iteration (3) may involve alternatives (2), and in turn, a sequence of instructions (1). The other way around, a sequence of instructions (1) may only appear as an iteration (3) through the mediation of alternatives (2). Note that an alternative may consist in a single choice, and an iteration a single cycle, degenerately.

A knowledge representation respecting the properties of meaningful interpretation must be able to comply with both types of dependency above and, conform the recursive nature of the Peircean categorical scheme, it must have the potential to be applied recursively. These conditions may put a great burden on a computational implementation of a Peircean knowledge representation.

Having introduced the basic properties of the three categories, we are ready to offer an informational analysis to the dependencies between them.

3.1 Informational analysis

In past research we have shown that, from Peirce's theory of categories, a knowledge representation can be derived [5]. This goal can be achieved in two ways: the first is, by offering an aspectual analysis to signs and assigning a process interpretation to the obtained hierarchy of sign aspects (see Fig. 1); the second is, through an informational analysis of phenomena. In [4] we have shown that the representations obtained by the two derivations can be isomorphic. By virtue of its more straightforward presentation, in this paper we will elaborate on the second alternative above.

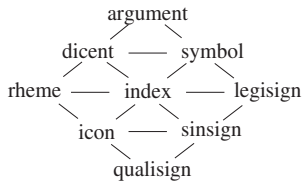

Fig. 1: A process interpretation of Peirce's hierarchy of sign aspects, introduced in [5]. Horizontal lines are used to designate interaction events between representations of the input from different perspectives (cf. sign aspects). The input of the process is associated with the qualisign position

Because thirdness can only be experienced (i.e. interpreted), perceived phenomena must be a thirdness. Following a theory of cognition [2], perceived phenomena must be an event representation of a change involved in the input interaction. Put differently, only if there is a change, an interaction may appear as an event. By virtue of the dependency between the three categories, perceived phenomena (cf. thirdness) involve a relation (cf. secondness), and in turn, a quality (cf. firstness). Below, in our analysis of phenomena we restrict ourselves to interactions between a pair of qualities, that we designate by $\mathtt{q}_2$ and $\mathtt{q}_1$. The term quality may refer to a single quality and a collection of qualities, ambiguously.

Qualities involved in an interaction must be independent, otherwise their co-occurrence may not involve a change hence an event. An interaction may be interpreted however, as a phenomenon of any category, potentially. From these conditions we may draw the conclusion that qualities involved in an interaction must convey information about their possible interpretation as a phenomenon of any one of the three categories.

In this paper we suggest that information involved in an interaction can be represented by a hierarchy of pairs of categorical information of qualities. See Fig. 2(a). An example is the pair (3,2), designating information enabling a meaningful (3) and a relational interpretation (2), involved in $\mathtt{q}_2$ and $\mathtt{q}_1$, respectively. In the domain of language processing, the type of information represented by (3,2) may correspond to information involved in the syntactic subject of a sentence, standing for an actually existent entity (cf. thirdness) and implicating (cf. secondness) the appearance of a characteristic property, represented by the predicate.

Following our informational analysis, in the next section we recapitulate a result from [5], and show how on the basis of a theory of cognitive activity a process can be derived which is isomorphic and analogous to the Peircean categorical representation depicted in Fig. 2(a). It is by virtue of *this* relation that the suggested process model can be called a Peircean model of KR.

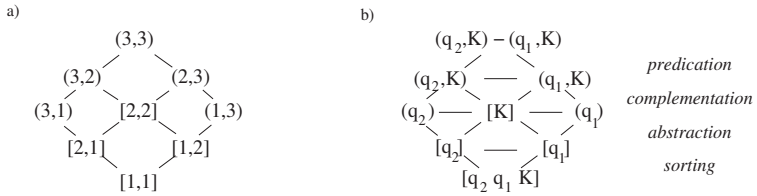

Fig. 2: (a) A hierarchical representation of information involved in an interaction between a pair of qualities, q_2 and q_1. A pair of integers is used to designate categorical information involved in q_2 and q_1 (in this order). (b) The process model of cognitive activity. Horizontal lines are used to designate interaction events between different input representations. The types of interpretation used are displayed on the right-hand side in italics

4 Process model

Following [2], we assume that the goal of cognitive activity is the generation of a response on the input stimulus. In a single interaction, the stimulus, appearing as an effect, is affecting the observer, occurring in some state. The qualities of this state (q_2) and effect (q_1), as well as memory knowledge (K) triggered by q_2 and q_1, form the input for information processing ($[\mathsf{q}_2\ \mathsf{q}_1\ \mathsf{K}]$). See Fig. 2(b). The occurring state (q_2) and effect qualities (q_1) are in the focus of the observer; the activated memory knowledge (K) is complementary.

From an informational stance, the goal of human processing is to establish a relation answering the question: why this effect is occurring to this state. In order to achieve this goal, the observer or interpreting system has to sort out the two types of qualities and context occurring in the input interaction ($[\mathsf{q}_2]$, $[\mathsf{q}_1]$, $[\mathsf{K}]$), abstract the type of qualities that are in focus into independent collections ((q_2), (q_1)), complete those collections with complementary knowledge by the interpreting system ((q_2,K), (q_1,K)), and through predication, merge the obtained representations into a single relation ((q_2,K)–(q_1,K)).

The isomorphism between the diagrams in Fig. 2 must be clear. An analogy between positions in the two diagrams can be explained as follows. The input, $[\mathsf{q}_2\ \mathsf{q}_1\ \mathsf{K}]$, expressing a potential for interpretation, corresponds to information represented by [1,1] (note that secondness and thirdness category information may be involved in [1,1], but that information is as yet not operational). The expressions obtained by sorting, $[\mathsf{q}_2]$, $[\mathsf{q}_1]$, and $[\mathsf{K}]$, exhibiting a potential for a relation involved in the input interaction, correspond to information represented by [2,1], [1,2] and [2,2]. For instance, [2,1] is an expression of relational information involved in q_2, and a potential for interpretation (e.g., as a relation) involved in q_1. An explanation of a relation between other positions in the two diagrams can be given analogously.

4.1 Limitations and potential of the model

Due to its computational character (cf. secondness), the model in Fig. 2(a) may not be able to represent triadic relations hence also meaningful interpretation (cf. thirdness). We may ask: can this model offer more than traditional theories of knowledge representation can?

In our view the answer can be positive. Through respecting the types of distinctions that can be signified by phenomena (cf. the nine positions in Fig. 1), the proposed theory may enable a systematic development of models of human processing. Due to a lack of a suitable ontology, traditional KR may not have this potential.

By virtue of the fundamental nature of categories, the process model, depicted in Fig. 2(b), may *uniformly* characterize human processing in any domain hence can be used as a meta-theory (and methodology) for KR as well. An advantage of a uniform representation of knowledge is its potential for merging information in different domains into a single representation by means of *structural coordination*, which can be more efficient than merging via translations between different representations. Experimental evidence for a uniform representation of information by the brain can be found in cognitive research by [1]. In this paper the authors show, by means of fMRI measurements, that language-related ('syntactic') and world-related ('semantic') knowledge processing can be quasi-simultaneous in the brain. Their results imply that human processing may not have sufficient time for a translation between representations in different knowledge domains (at least, in the domains tested) hence the use of a uniform representation could be inevitably necessary.

Illustrations of the theoretical potential of the proposed model of KR in various domains, including natural language processing, reasoning and mathematical conceptualization, can be found in [5].

References

1. P. Hagoort, L. Hald, M. Bastiaansen, and K-M. Petersson. Integration of word meaning and world knowledge in language comprehension. *Science*, 304:438–441, 2004.
2. S. Harnad. *Categorical Perception: The groundwork of cognition.* Cambridge University Press, Cambridge, 1987.
3. C.S. Peirce. *Collected Papers of Charles Sanders Peirce.* Harvard University Press, Cambridge, 1932.
4. J.J. Sarbo and J.I. Farkas. Towards meaningful information processing: A unifying representation for Peirce's sign types. *Signs – International Journal of Semiotics*, 7:1–41, 2013.
5. J.J. Sarbo, J.I. Farkas, and A.J.J. van Breemen. *Knowledge in Formation: A Computational Theory of Interpretation.* Springer (eBook: http://dx.doi.org/10.1007/978-3-642-17089-8), Berlin, 2011.
6. J. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3:417–424, 1980.

Linguistic Affordances: Making sense of Word Senses

Alice Ruggeri and Luigi Di Caro

Department of Computer Science, University of Turin
Corso Svizzera 185, Torino, Italy
{ruggeri,dicaro}@di.unito.it

Abstract. In this position paper we want to focus the attention on the roles of word senses in standard Natural Language Understanding tasks. We first identify the main problems of having such a rigourous and inflexible way of discriminating among different meanings at word-level. In fact, in human cognition, we know the process of language understanding refers to a more shaded procedure. For this reason, we propose the concept of *linguistic affordances*, i.e., combinations of objects properties that are involved in specific actions and that help the comprehension of the whole scene being described. The idea is that similar verbs involving similar properties of the arguments may refer to comparable mental scenes. This architecture produces a converging framework where meaning becomes a distributed property between actions and objects, without having to differentiate among terms and relative word senses. We hope that this contribution will stimulate the debate about the actual effectiveness of current Word Sense Disambiguation systems towards more cognitive approaches able to go beyond word-level automatic understanding of natural language.

1 Background

In linguistics, a *word sense* is the meaning ascribed to a word in a given context. A single word can have multiple senses. For instance, within the well-known lexical database WordNet [1], the word "play" has 35 verb senses and 17 senses as noun. This phenomenon is called polysemy. However, this must be distincted from the concept of homonymy, where words share the same spelling and the same pronunciation, having different and unrelated meanings. According to the human process of disambiguating meanings, the man reads a word at a time through a process called "word sense disambiguation". In the Natural Language Processing field, there exist numerous systems to automate this task, relying on existing ontologies [2, 3] rather than through statistical approaches [4, 5].

From another perspective, an *affordance* is linked to the meaning of an action that is dynamically created by the interaction of the involved agents [6–8]. Dropping this principle in natural language, an action (for example indicated by the use of a verbal phrase) will have a certain meaning that is given by the interaction between the agent and the receiver, and more particularly by the

their properties. The idea is that *different* combinations of subjects and objects with their properties are likely to lead to *different* actions in terms of execution, or final outcome.

2 Linguistic Affordances

In this work, we want to focus on the application of the concept of *affordance* to the natural language understanding made by machines. If a computer could comprehend language meanings at a more cognitive level, it would allow more complex and fine-grained automatic operations, leading to highly-powerful systems for Information Extraction [9] and Question Answering [10].

The meaning of a word is a concept that is very easy to understand by humans. A word sense is directly tied to a single entry in a dictionary. It applies to nouns and verbs in the same way. Given a word with more than one meaning, humans must proceed with the process of disambiguation using all the available information coming from the context of use.

The affordance of a word is a more complex thing. First of all, the word in question must refer to an action. For this reason, it is particularly oriented towards verbs (although other language constructions can refer to actions or events). In addition, the affordances related to an action (we will now use the more precise term "action" instead of "word") is suggested by the properties (also called qualities, attributes, or characteristics) of those who act and those who receive the action, together. The affordance is more tied with the cognitive aspect of an action rather than its encyclopedic meaning. More precisely, it refers to *how the action can be mentally imagined*. This is also in line with [11, 12], i.e., meanings are relativized to scenes.

For these reasons, the affordance is not directly linked to an entry in a dictionary. It has no direct link with a descriptive meaning. Nevertheless, it can coincide with it. In general, affordances and meanings are two distinct concepts that travel on separate tracks, but which can also converge on identical units. On the contrary, it may be the case that two distinct senses for a verb accurately reflect two different subject-object contexts. In this case, word senses and affordances coincide.

Still, a single sense can include multiple affordances. It is the case where a word with a single meaning can be applied to multiple subject-object combinations, creating different mental images of the same action.

Finally, two distinct word senses could not theoretically lead to a single linguistic mental image of an action, since two different meanings are likely to identify two different mental images. We think that it would be interesting to see how much of such theoretical concept can be considered valid. Potentially, two word senses can be very close semantically, inducing to a single mental image (and therefore a single combination of properties). This, undoubtedly, also depends on the level of granularity that has been chosen during the creation of the possible senses related to a word. In any case, we want to stress the actual independence between the two perspectives.

Word senses are completely separated. This means that they refer to meanings that have the same degree of semantic distance between them. However, this results to be quite approximate, since human cognition does not work this way. A sense "x" can be very similar to another sense "y", while very distant from a third one "z". More in detail, there exist the concept of "similarity between senses" thought as the similarity of the mental models that they generate [13]. These abstractions are plausibly created by combining the properties of the agents that are involved in the action, thus through the affordances that they exhibit.

Let us think at the WordNet entry for the verb "to play". Among all 35 word senses, there exist groups that share some semantics. For instance, the word sense #3 and the word senses #6 and #7 are defined by the following descriptions:

- To play #3: play on an instrument ("the band played all night long")
- To play #6: replay as a melody ("play it again, Sam", "she played the third movement very beautifully")
- To play #7: perform music on a musical instrument ("he plays the flute", "can you play on this old recorder?")

It is noticeable that the three word senses refer to similar meanings. Within the WordNet knowledge base, the lexicographers have manually grouped word senses according to this idea. However, coverage of verb groups is incomplete. Moreover, having groups of senses only solves the semantic similarity problem to a limited extent, since the concept of similarity usually deals with more fine-grained analyses. In literature, there are several computational models to classify words of a text into relative word senses. On the contrary, there are no computational models to identify "scenes" or "mental images" in texts.

The Word Sense Disambiguation task is one of the most studied in computational linguistics for several reasons:

- there are a lot of available resources (often manually produced) presenting dictionaries and corpus annotated with word senses (such as WordNet and the SemEval competition series [1, 14]).
- it has a significant impact in the understanding of language from the computational point of view. Through the disambiguation of terms in texts it is possible to increase the level of accuracy of different systems for Information Retrieval, Information Extraction, Text Classification, Question Answering, and so on.

The extraction of linguistic affordances in texts is an issue rather untouched, for different (but correlated) reasons:

- there are no resources and manual annotation of this type of information
- affordances have a more cognitive aspect than word senses, thus they seem less applicable

Nevertheless, we think that this type of analysis can represent a significant step forward on the current state of the art.

3 Conclusions

In this paper we presented the limits of having fixed and word-level semantic representations, i.e., word senses, for automatic tasks like Information Extraction and Semantic Search. Instead, we proposed an orthogonal approach where meaning becomes a distributed property between verbs and arguments. In future work we aim at studying how arguments properties distribute over actions indicated by specific verbs in order to test the idea, making first comparisons with standard word sense-based approaches for automatic natural language understanding.

References

1. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11) (1995) 39–41
2. Agirre, E., Martinez, D.: Knowledge sources for word sense disambiguation. In: Text, Speech and Dialogue, Springer (2001) 1–10
3. Curtis, J., Cabral, J., Baxter, D.: On the application of the cyc ontology to word sense disambiguation. In: FLAIRS Conference. (2006) 652–657
4. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2009) 33–41
5. Navigli, R., Lapata, M.: An experimental study of graph connectivity for unsupervised word sense disambiguation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(4) (2010) 678–692
6. Gibson, J.: The concept of affordances. Perceiving, acting, and knowing (1977) 67–82
7. Ortmann, J., Kuhn, W.: Affordances as qualities. In: Formal Ontology in Information Systems Proceedings of the Sixth International Conference (FOIS 2010). Volume 209. (2010) 117–130
8. Osborne, F., Ruggeri, A.: A prismatic cognitive layout for adapting ontologies. In: User Modeling, Adaptation, and Personalization. Springer (2013) 359–362
9. Sarawagi, S.: Information extraction. Foundations and trends in databases **1**(3) (2008) 261–377
10. Mendes, A.C., Coheur, L.: When the answer comes into question in question-answering: survey and open issues. Natural Language Engineering **19**(1) (2013) 1–32
11. Fillmore, C.J.: Frame semantics and the nature of language*. Annals of the New York Academy of Sciences **280**(1) (1976) 20–32
12. Fillmore, C.: Frame semantics. Linguistics in the morning calm (1982) 111–137
13. Johnson-Laird, P.: Mental models: Towards a cognitive science of language, inference, and consciousness. Number 6. Harvard University Press (1983)
14. Agirre, E., Diab, M., Cer, D., Gonzalez-Agirre, A.: Semeval-2012 task 6: A pilot on semantic textual similarity. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, Association for Computational Linguistics (2012) 385–393

Towards a Formalization of Mental Model Reasoning for Syllogistic Fragments

Yutaro Sugimoto[1], Yuri Sato[2] and Shigeyuki Nakayama[1]

[1] Department of Philosophy, Keio University, Tokyo, Japan
[2] Interfaculty Initiative in Information Studies, The University of Tokyo, Japan
{sugimoto,nakayama}@abelard.flet.keio.ac.jp,sato@iii.u-tokyo.ac.jp

Abstract. In this study, Johnson-Laird and his colleagues' mental model reasoning is formally analyzed as a non-sentential reasoning. Based on the recent developments in implementations of mental model theory, we formulate a mental model reasoning for syllogistic fragments in a way satisfying the requirement of formal specification such as mental model definition.

1 Introduction

Recently, non-sentential or diagrammatic reasoning has been the subject of logical formalization, where diagrammatic reasoning is formalized in the same way as sentential reasoning is formalized in modern logic (e.g., [11]). In line with the formal studies of diagrammatic logic, we present a formalization of *mental model* reasoning, which was introduced by [6], as a cognitive system of reasoning based on non-sentential forms.

The mental model theory has been about cognitive-psychological theory, providing predictions of human performances and explanations of cognitive processes. Meanwhile, the theory has been attracted attention from various research fields including AI, logic, and philosophy beyond the original field (e.g., [2, 5, 8]) considering it can be taken as an applied theory based on the mathematical and logical notions such as models and semantics. It has been discussed not only empirical plausibility but also a formal specification of mental model reasoning.

The problem we focus on is that the definition of "mental model" is not provided properly within the explanations of mental model theory. It is a key to understand the full system and a step to give formal specifications of the theory. Recently, Johnson-Laird and his colleagues' several implementation works were made public[1], revealing the detailed procedures of the theory. However formal specifications or definitions requested here are still not included in their programs. An appropriate way to address the problem is to formulate the theory in accordance with their programs satisfying the requirements of the formal specification such as mental model definition. Our view is consistent with the seminal study in [1], who took the first step towards formalization of mental model reasoning while presenting a computer programs of it.

The theory was originally formulated for categorical syllogisms [6], therefore we begin our formalization project in the domain of syllogisms only. Particularly, we focus on the more recent version in [3] and the corresponding computer program [9].

[1] See their laboratory's webpage: http://mentalmodels.princeton.edu/programs

Before the formal work, we provide a brief overview of mental model theory with its illustrations of solving processes of syllogistic reasoning. The basic idea underlying the mental model theory is that people interpret sentences by constructing mental models corresponding to situations and make inferences by constructing counter-models. Mental models consist of a finite number of tokens, denoting the properties of individuals. For example, the sentence, "*All A are B*," has a model illustrated on the leftmost side of Fig.1, where each row represents an individual. Here, a row consisting of two

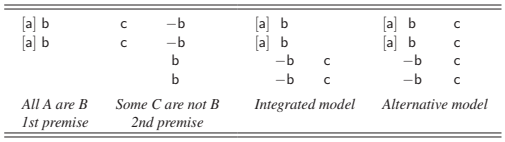

Fig. 1: Solving processes in mental model theory for a syllogistic task.

tokens, a and b, refers to an individual which is *A* and *B*. Furthermore, the tokens with square brackets, [a], express that the set containing them is exhaustively represented by these tokens and that no new tokens can be added to it. By contrast, a sequence of tokens without square brackets can be extended with new tokens so that an alternative model is constructed. However, such an alternative model is not taken since *parsimonious descriptions* are postulated to be preferred (chap. 9 of [7]). In a similar way, the sentence "*Some C are not B*" has a model illustrated on the second from the left of Fig.1. Here, a row having a single token, b, refers to an individual which is *B* but not *C*. Furthermore, the same thing can be also represented by the use of the device of "–" denoting negation. A row consisting of two tokens, c and –b, refers to an individual which is *C* but not *B*.

The right side of Fig. 1 shows a model integration process with these two premises. In this process, the two models in the left side of Fig.1 are integrated into a single model by identifying the tokens of set *B*. After the integration process, a searching process for counterexamples is performed, and alternative models are constructed. In this case of Fig.1, an alternative models is constructed from the integrated model by adding new tokens (i.e., token c). Since each tokens of set *A* are corresponding to tokens of set *C*, one of tentative conclusions "*Some A are not C*" is refuted. Hence, this tentative conclusion can be considered a *default assumption*, i.e., it can be specified as a conclusion by default and it can be revised later if necessary (chap. 9 of [7]). Instead, by observing that some tokens of set *C* are disjoint from the tokens of set *A*, one can extract a valid conclusion "*Some C are not A*" from the alternative model.

In the next section, we provide a formalization of mental model theory including the features: *parsimonious descriptions* and *default assumption*. We note here that our work does not intend to provide a normative and sophisticated version of mental model theory. Hence our work is not in line with the stance as taken in [4, 5], where the features above, postulated in mental model theory, are less focused.

2 A Mental Model Reasoning System

We provide a formalization for a mental model (syllogistic) reasoning system. Since the prototype program [9] is fully implemented by Common Lisp, it lacks static type infor-

mation [12] and mental models are not defined explicitly. In order to treat the system formally, types serve significant role. Firstly, we describe the system as a finite state transition machine and provide type information to main procedures. Fig.4 shows the transitions from one state (model) to another by following processes: (1) constructing mental models of premises, (2) integrating premise models into an initial model, (3) drawing a tentative conclusion from an initial model, (4) constructing alternative model by falsification, and (5) responding a final conclusion.

2.1 Mental Model Construction

Though actual mental models are constructed implicitly in human cognition, the computational (syntactical) representation for mental models is constructed explicitly by the interpreter which converts semi-natural syllogistic language into computational representation for mental models. Accordingly, we first define a formal language for (semi-natural) syllogistic language by extended BNF following [9]. See Fig.2.

⟨*sentence*⟩ ::= ⟨*np*⟩ ⟨*pred*⟩
| ⟨*np*⟩ ⟨*negpred*⟩
| ⟨*neg-np*⟩ ⟨*pred*⟩
⟨*np*⟩ ::= ⟨*quant*⟩ ⟨*term*⟩
⟨*neg-np*⟩ ::= ⟨*neg-quant*⟩ ⟨*term*⟩
⟨*pred*⟩ ::= ⟨*cop*⟩ ⟨*term*⟩
⟨*negpred*⟩ ::= ⟨*cop*⟩ ⟨*neg*⟩ ⟨*term*⟩
⟨*term*⟩ ::= A | B | C
⟨*quant*⟩ ::= All | Some
⟨*neg-quant*⟩ ::= No
⟨*neg*⟩ ::= not
⟨*cop*⟩ ::= are

Fig. 2: Grammar for syllogistic language

⟨*token*⟩ ::= ⟨*atom*⟩
| ⟨*lsqbracket*⟩ ⟨*atom*⟩ ⟨*rsqbracket*⟩
| ⟨*neg*⟩ ⟨*atom*⟩
| ⟨*nil*⟩
⟨*lsqbracket*⟩ ::= [
⟨*rsqbracket*⟩ ::=]
⟨*atom*⟩ ::= a | b | c
⟨*neg*⟩ ::= −
⟨*nil*⟩ ::=

Fig. 3: Grammar for mental model tokens

Next we give a definition for mental model units for syllogistic reasoning as follows: A *mental model* is a *class of models*[2] s.t. $m \times n$ matrix of *tokens* where $m \geq 2$ and $3 \geq n \geq 1$. A *row* or an *individual* of a mental model is a finite array of tokens (*model*) where each atoms occur at most once. A *column* or a (*property*) of a mental model is a finite array of tokens where tokens contain any different atoms cannot co-occur. If square bracketed tokens occur in a column, only negative atoms can be added. Fig. 3 is the vocabulary and grammar for mental model *token*s. Since the detail of language translation is not our current concern, we do not give a specification for the language interpreter[3]. Alternatively, we give examples of translations. Let X,Y denote terms A, B, C. The four types of syllogistic sentences can be translated to mental models as follows:

All X are Y	Some X are Y	No X are Y	Some X are not Y
⇓	⇓	⇓	⇓
[x] y	x y	[x] −y	x −y
[x] y	x	[x] −y	x −y
	y	[y]	y
		[y]	y

[2] For a treatment of a mental model as a class of models, see [2].
[3] For the detail of typical formal language transformation processes, see e.g. [10].

2.2 Integrating Premises into Initial Model

We give a description for the integration process of premises into an initial model via mid-term tokens (Fig.5). The integration process can be considered nearly as having functional type $f : P \to P \to M$ (P is a type of premisses: P_1, P_2).

Reordering and Switching Since syllogisms have several "figures" according to the order of premises and term arrangements, the actual integration procedure should occur after reordering terms and switching premises as preprocesses. This preprocess has the following four patterns:

(1) If the term order of P_1 is AB and P_2 is BC, nothing happens.
(2) If the term order of P_1 is BA and P_2 is CB, starts with P_2.
(3) If the term order of P_1 is AB and P_2 is CB, swaps second model round and adds it.
(4) If the term order of P_1 is BA and P_2 is BC, swaps first model then adds second model.

Finding a middle atom The procedure of *finding a middle atom: a* can be considered as having a functional type $g : P \to P \to a$. The actual implementation for this is a similar to set intersection operation for the affirmative tokens (tokens which do not contain negatives). For example, when two premises are as Fig.1, $\{a,a,b,b\} \cap \{c,c,b,b\} = b$.

Match The procedure of matching premises P_1, P_2, and middle atom a could have functional type $rec : P \to P \to a \to M$. This recursive procedure calls `join` as sub procedure to join the premises to an integrated model.

Join This recursive procedure takes a mid atom and two individuals, and joins two individuals together setting new mid to exhausted if one or other was exhausted in first individual or second individual. This procedure could have a recursive functional type: $rec : a \to Indiv \to Indiv \to Indiv$.

2.3 Drawing a Conclusion from a Model

Drawing a conclusion (Fig.6) is a procedure which takes an integrated (initial) model and dispatches whether it contains negative token or not. It then dispatches further based on the predicates (all-isa, some-isa, no-isa, and some-not-isa) and returns *corresponding answers*.[4] If the predicates return #f, then it returns "No Valid Conclusion." The followings are sub procedures of `conclude`:

all-isa takes a model which has end terms X, Y and returns the answer "All X are Y" iff all subjects are objects in individuals in model. This has a functional type: *all-isa* : $M \to A$. For example, if a model M : $\begin{smallmatrix}[a] & b & c\\ [a] & b & c\end{smallmatrix}$ is given, where end terms are A and C, then returns the answer "All A are C."

some-isa takes a model which has end terms X, Y and returns the answer "Some X are Y" iff at least one individual in model contains positive occurrences of both subject and object atoms. This has a functional type: *some-isa* : $M \to A$. For example, if a model : $\begin{smallmatrix}[a] & [b] & c\\ [a] & [b] & \\ & c & \end{smallmatrix}$ is given when end terms are A and C then returns the answer "Some A are C".

[4] Notice: since possible conclusions have term order: Subj-Obj and Obj-Subj, `conclude` is executed twice respectively. For simplicity, we omit the second execution of `conclude`.

no-isa takes a model which has end terms X, Y and returns "No X are Y" iff no subject end term is object end term in any individuals in model. This has a functional type: *no-isa* : $M \rightarrow A$. For example, if a model M : $\begin{smallmatrix}[a]\ -b\\ [a]\ -b\\ \\ [b]\ [c]\\ [b]\ [c]\end{smallmatrix}$ is given when end terms are A and C then returns the answer "No A are C."

some-not-isa takes a model which has end terms X, Y and returns "Some X are not Y" iff at least one subject occurs in individuals without object. This has a functional type: *some-not-isa*: $M \rightarrow A$. For example, if a model M : $\begin{smallmatrix}[a]\ b\ c\\ [a]\ b\ c\\ -b\ c\\ -b\ c\end{smallmatrix}$ is given when end terms are A and C then returns the answer "Some A are not C."

2.4 Constructing Alternative Model

Once the mental model theory constructs an initial model and draws a tentative conclusion, the theory, according to the rules, tries to construct an alternative model in order to refute the conclusion (i.e., default assumption). The process of falsification (Fig.7) takes a model and dispatches whether it contains negative token or not. Then based on the predicates (breaks, add-affirmative, moves, and add-negative) it tries to modify the model. If succeeded, returns an alternative model and call `conclude` again. If failed, the recursive call of this procedure terminates. Here are main constructs of `falsify`:

breaks has a functional type: *breaks* : $M_1 \rightarrow M_2$. *breaks* finds an individual containing two end terms with non-exhaustive mid terms, divides it into two, then returns new (broken) model or returns *nil*. For example, if M_1 is $\begin{smallmatrix}a\ b\ c\end{smallmatrix}$, then *breaks*: $\begin{smallmatrix}a\ b\ c\end{smallmatrix} \rightarrow \begin{smallmatrix}a\ b\\ \ \ b\ c\end{smallmatrix}$.

add-affirmative has a functional type: $add^+ : M_1 \rightarrow M_2$. If add^+ succeeds, then it returns a new model $M_2$with added item (added model), else it returns *nil* if conclusion is not A-type ("All X are Y") or if there is no addable subject item.

For example, if M_1 is $\begin{smallmatrix}[a]\ [b]\ c\\ [a]\ [b]\ c\end{smallmatrix}$, then add^+: $\begin{smallmatrix}[a]\ [b]\ c\\ [a]\ [b]\ c\end{smallmatrix} \rightarrow \begin{smallmatrix}[a]\ [b]\ c\\ [a]\ [b]\ c\\ \ \ \ \ c\end{smallmatrix}$.

moves has a functional type: *moves* : $M_1 \rightarrow M_2$. If there are exhausted end items not connected to other end items or their negs (i.e E-type ("No X are Y") conclusion), and if the other end items are exhausted or O-type ("Some X are not Y") conclusion, then it joins them. Otherwise joins one of each and returns *nil* if the first end item cannot be moved even if a second one can be.

E.g., if M_1 is $\begin{smallmatrix}[a]\ -b\\ [a]\ -b\\ \\ [b]\ -c\\ [b]\ -c\\ \\ [c]\\ [c]\end{smallmatrix}$, then *moves*: $\begin{smallmatrix}[a]\ -b\\ [a]\ -b\\ \\ [b]\ -c\\ [b]\ -c\\ \\ [c]\\ [c]\end{smallmatrix} \rightarrow \begin{smallmatrix}[a]\ -b\ [c]\\ [a]\ -b\ [c]\\ \\ [b]\ -c\\ [b]\ -c\end{smallmatrix}$. When this procedure is called by `falsify`, `neg-braking` (similar procedure to `breaks`) is also called as an argument.

add-negative has functional type: $add^- : M_1 \rightarrow M_2$. It returns a new model with added item (add-neged model), or returns *nil* if conclusion is not O-type or if there is no addable subject item. E.g., if M_1 is $\begin{smallmatrix}[a]\ b\\ [a]\ b\\ -b\ c\\ -b\ c\end{smallmatrix}$, then add^-: $\begin{smallmatrix}[a]\ b\\ [a]\ b\\ -b\ c\\ -b\ c\end{smallmatrix} \rightarrow \begin{smallmatrix}[a]\ b\ c\\ [a]\ b\ c\\ -b\ c\\ -b\ c\end{smallmatrix}$.

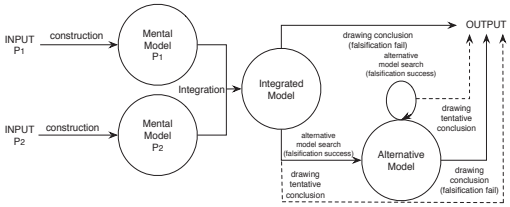

Fig. 4: Finite state transition machine diagram for syllogisms

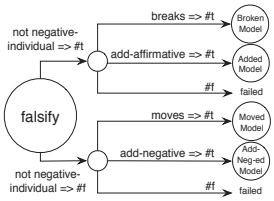

Fig. 7: Falsification process

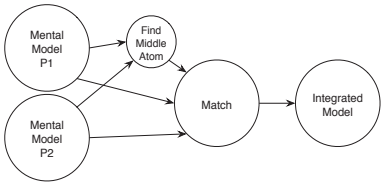

Fig. 5: Integration process

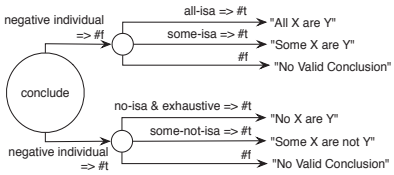

Fig. 6: Drawing conclusion process

Acknowledgements. This study was supported by MEXT-Supported Program for the Strategic Research Foundation at Private Universities (2012-2014) and Grant-in-Aid for JSPS Fellows (25·2291).

References

1. Bara, B.G., Bucciarelli, M., & Lombardo, V. (2001). Model theory of deduction: A unified computational approach. *Cognitive Science*, 25, 839–901.
2. Barwise, J. (1993). Everyday reasoning and logical inference. *Behavioral and Brain Sciences*, 16, 337–338.
3. Bucciarelli, M. & Johnson-Laird, P.N. (1999). Strategies in syllogistic reasoning. *Cognitive Science*. 23, 247–303.
4. Clark, M. (2010). *Cognitive Illusions and the Lying Machine*. Ph.D Thesis, Rensselaer P.I.
5. Hintikka, J. (1987). Mental models, semantical games, and varieties of intelligence. In *Matters of Intelligence* (pp. 197–215), Dordrecht: D. Reidel.
6. Johnson-Laird, P.N. (1983). *Mental Models*. Cambridge, MA: Harvard University Press.
7. Johnson-Laird, P.N., & Byrne, R. (1991). *Deduction*. Hillsdale, NJ: Erlbaum.
8. Lowe, E.J. (1993). Rationality, deduction and mental models. In *Rationality* (pp. 211–230), Taylor & Frances/Routledge.
9. Mental Models and Reasoning Lab. Syllogistic reasoning code [Computer program]. Retrieved Oct.10, 2012, from `http://mentalmodels.princeton.edu/programs/Syllog-Public.lisp`.
10. Mitchell, J.C. (1996). *Foundations for Programming Languages*. Cambridge, MIT Press.
11. Shin, S.-J.(1994). *The Logical Status of Diagrams*. New York: Cambridge University Press.
12. Steele, Jr. G.L. (1990). *Common LISP: The Language (2nd ed.)*. Newton, MA: Digital Press.