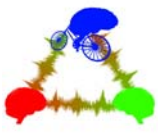
Language and Robots

Proceedings of the Symposium

10-12 December 2007, Aveiro, Portugal



Title: *Language and Robots: Proceedings of the Symposium. 10-12 December 2007, Aveiro, Portugal*

Editors: Luís Seabra Lopes, Tony Belpaeme, Stephen J. Cowley

Cover: traditional boats in Aveiro lagoon, aerial view of Aveiro and some local robots: Carl, CAMBADA soccer team, humanoid robot, ROTA autonomous car.

Graphical design of cover, printing and binding: DESIGNEED - DESIGN E PUBLICIDADE, UNIP. LDA.

ISBN: 978-972-96895-2-9

Depósito legal: 268208 / 07

Symposium on Language and Robots

10-12 December 2007

Aveiro, Portugal

Welcome to the Symposium on Language and Robots! Welcome to the University of Aveiro!

The symposium aims to explore synergies and identify areas of collaboration between robotics and the language sciences. As starting point for the discussions, a perspective is proposed in which language is seen as a dynamic and distributed cognitive process.

The origins, evolution and acquisition of language and its role in human societies have long been studied by philosophers, linguists, psychologists, neuroscientists and cognitive scientists. In recent years, a distributed view of cognition and language has emerged. Control of embodied action is now seen as an emergent property of a distributed system composed of brain, body and environment. Language ceases to be seen as a formal underlying system and, instead, becomes a heterogeneous set of culturally distributed processes. Language is a cultural product, perpetually open-ended and incomplete, and partly ambiguous. Both learning to talk and language evolution involve not only internal, but also cultural, social and affective processes.

In this context, many research questions open up: How does language transform human cognitive processes? How is language grounded in perception and action? In what ways does human phenomenology depend on linguistic experience? Can a distributed perspective on language clarify the nature of silent rehearsal (internal thought processes)? How does this relate to consciousness? How is language used to achieve joint experience? What is the embodied basis of cognition and social semiosis?

While the language sciences have, until now, focused on language in human societies, the robotics and artificial intelligence communities are increasingly developing user-friendly robots, that is, robots that are flexible, adaptable and easy to command and instruct. These artificial agents need to cognitively interpret perception and action, accumulate and manipulate semantic information for decision-making and interact with human subjects using natural language.

There are two obvious contact areas between robotics and the language sciences. In the first, robots can be used as simulation models for the empirical study of language origins, evolution and acquisition. This is an extension of the computational modeling approach to language. In the second area, current knowledge about language as a cultural product can be used to design and develop robots for practical applications.

The symposium is therefore intended for exploring the following issues:

- How can robots ground and use language for practical applications?
- How can robots be used for empirical work in the language sciences?
- How can robots acquire language when language is distributed?
- What does robotics imply for the language sciences?
- What questions do roboticists want to ask the language sciences?

978-972-96895-2-9

- How can the language sciences contribute to the theoretical and practical study of how humans interact with robots?
- How can the language sciences evolve to address societies that include robots?

An open call for papers invited papers to address these issues, welcoming speakers from a wide range of disciplines and backgrounds. The proceedings include a total of 22 high quality contributions submitted to the open call. Due to the high number of good quality papers, some of them could not be accepted for oral presentation, so authors were invited to present them in a poster session. The program thus includes 17 oral presentations and 5 poster presentations. The symposium format also includes 5 Invited Talks, by prominent researchers in robotics, language sciences and related fields. The whole event will extend to two and half days.

After the symposium, participants will be invited to submit a paper to a special issue of the journal "Connection Science" on language and robots.

The workshop is kindly sponsored by the *euCognition* network (*European Network for the Advancement of Artificial Cognitive Systems*) and by *FCT* (*Fundação para a Ciência e a Tecnologia*, Lisbon, Portugal). We thank and appreciate very much the collaboration of Nautília Maia, from the Department of Electronics, Telecommunications and Informatics of the University of Aveiro, in managing registrations and some other local arrangements.

We hope you enjoy the workshop, and trust it will be a highly productive and sociable event.

Luís Seabra Lopes
Tony Belpaeme
Stephen J. Cowley

Proc. LangRo'2007
978-972-96895-2-9

Table of Contents

POSTER PRESENTATIONS

Proc. LangRo'2007
978-972-96895-2-9

Circuit sharing for action-grounded meaning

Michael L. Anderson

Department of Psychology, Franklin & Marshall College, Lancaster, PA USA
and
Institute for Advanced Computer Studies, University of Maryland, College Park, MD USA
michael.anderson@fandm.edu

Extended Abstract

In this talk I will review some of the interesting behavioral evidence that human language and motor systems are deeply intertwined—such as the action sentence compatibility effect (Glenberg & Kaschak, 2002); linguistic and spatio-motor disfluency effects (Casasanto & Lozano, 2007); and various conceptual-motor simulation theories (Barsalou, 1999; Richardson, et al., 2003). I will then try to place these findings in the context of a novel theory of the evolution and organization of the human cortex, the massive redeployment hypothesis (Anderson, 2007a; 2007b).

The massive redeployment hypothesis (MRH) is a theory about the functional organization of the human cortex, offering a middle course between strict localization on the one hand, and holism on the other. Central to MRH is the claim that cognitive evolution proceeded in a way analogous to component reuse in software engineering, whereby existing components—originally developed to serve some specific purpose—were used for new purposes and combined to support new capacities, without disrupting their participation in existing programs. If the evolution of cognition was indeed driven by such exaptation, then we should be able to make some specific empirical predictions regarding the resulting functional topography of the brain. This talk outlines three such predictions, and some of the evidence supporting them. First, any given brain area is typically redeployed in support of many cognitive functions, and such redeployment will not respect traditional domain boundaries (that is, brain areas are not domain-restricted entities) Second, more recently evolved cognitive functions will utilize more, and more widely scattered brain areas, and third, evolutionarily older brain areas will be deployed in more cognitive functions.

Having provided some support for the plausibility of MRH as a general theory of cortical organization, my suggestion will be that the best way to understand the apparent interrelations between language and motor control is in terms of the activation of shared brain regions ***not*** due to real-time perceptual-motor simulations of conceptual structures, but rather due to the literal sharing of the same neural circuits by different functional complexes. This, I will argue, has significant implications for our understanding of both motor control and of language, and for what it might take to build "meaning machines", implications somewhat different from those that have generally been taken to follow from the prevailing simulation-based theories.

For instance, something of the character of the perceptual and motor systems is evident in the language understanding system—putting linguistic elements together in a meaningful sentence is like putting motor primitives together in an executable motor plan.

Moreover, there is a reverse implication that is worth considering: what does the fact that language is built in part on motor-control circuits tell us about *motor control*? Since affordances, the perceived availability of objects for certain kinds of interaction, aren't just motor programs, but features of the environment with specific significance for the organism, this opens the possibility that the motor control system is also, already, a primitive meaning processor (Gorniak & Roy, 2006). This would offer one explanation of how it is even possible to leverage motor control to support and constrain higher-order processes like language understanding. After all, on a more mechanistic understanding of the nature of motor control, it would be nearly impossible to say why a motor-control system would have any of the right basic elements for building a language understanding system.

References

Anderson, M.L. (2007a). Evolution of cognitive function via redeployment of brain areas. *The Neuroscientist, 13*(1): 13-21.

Anderson, M.L. (2007b). The massive redeployment hypothesis and the functional topography of the brain. *Philosophical Psychology, 21*(2): 143-174.

Barsalou, L. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577-609.

Casasanto, D. & Lozano, S. (2007). Meaning and Motor Action. *Proceedings of 29th Annual Conference of the Cognitive Science Society* (pp. 149-154). Hillsdale, NJ: Lawrence Erlbaum Associates.

Glenberg, A. M., & Kaschak, M. P. (2002). Grounding language in action. *Psychonomic Bulletin and Review, 9*, 558-565.

Gorniak, P. and Roy, D. (2006). Perceived affordances as a substrate for linguistic concepts. *Proceedings of the twenty-eighth annual meeting of the Cognitive Science Society*.

Richardson, D., Spivey, M., Barsalou, L., & McRae, K. (2003). Spatial representations activated during real-time comprehension of verbs. *Cognitive Science, 27,* 767-780.

Towards a Mechanistic Model of Referential Semantics

Deb Roy

MIT Media Lab
Massachusetts Institute of Technology
Cambridge, MA,USA

dkroy@media.mit.edu

Abstract

Based on experiments in building conversational robots (Roy 2003; Roy et al. 2004; Hsiao, 2007), a new framework for grounding semantics in sensory-motor interaction is emerging (Roy 2005; Roy in press). In this talk I will focus on a mechanistic (physical-computational) model of reference within this framework. The basic building blocks of the model are interactive sensory-motor processes which couple an embodied system's external physical environment to its internal information structures and processes. The presence of an instantiated interactive process within the system is treated as a persistent representational element that stands for the environmental object to which it is actively coupled. The translation of sensory-motor activity to stable internal representational elements provides the basis for symbolic communication.

References

Deb Roy. (2003). Grounded Spoken Language Acquisition: Experiments in Word Learning. *IEEE Transactions on Multimedia*, 5(2): 197-209.

Deb Roy, Kai-Yuh Hsiao, and Nikolaos Mavridis. (2004). Mental Imagery for a Conversational Robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Volume 34 , Issue 3, pages 1374-1383.

Kai-yuh Hsiao. (2007) *Embodied Object Schemas for Grounding Language Use*. Ph.D. in Media Arts and Sciences, Massachusetts Institute of Technology.

Deb Roy. (2005). Semiotic Schemas: A Framework for Grounding Language in the Action and Perception. *Artificial Intelligence*, 167(1-2): 170-205.

Deb Roy. (in press). A Mechanistic Model of Three Facets of Meaning. Chapter to appear in *Symbols, Embodiment, and Meaning*, de Vega, Glenberg, and Graesser, eds.

Proc. LangRo'2007
978-972-96895-2-9

Towards Learning by Interacting

Gerhard Sagerer

Bielefeld University, Germany
sagerer@techfak.uni-bielefeld.de

Abstract

In natural tutoring situations, recent developmental research has revealed that the way knowledge is transferred differs from the implicit traditional assumption of robotics within learning by demonstrating scenarios. While learning by watching implied analyses of the ongoing situation, a modified situation is presented to infants. For example, when talking to infants, adults modify their speech, known as Motherese. Regarding other modalities, it has been suggested that when introducing novel words for objects and their functions, parents move differently towards children (i.e. they move new objects in temporal synchrony to the novel label) than towards adults. Analyzing these behavior modifications using objective measurement techniques, we showed that behavior in parents (fathers and mothers) is modified multimodally also for familiar actions.

What advantages does this behavior bring for robots? We know so far that external real-world situations are rich and complex. Maybe far too rich to be captured accurately by a robot's internal models. Infants do not have to cope with the complexity by their own. Instead, the way adults present the world to them seems to facilitate both information registration and encoding. Simulating child's attention, we noticed that modified actions can help infants to detect the initial and goal states of actions. Thus, infants do not have to discover the meaning of a movement by themselves. Instead, adults seem to provide some structure.

Fluid Language Games and the Emergence of Grammar

Luc Steels

University of Brussels (VUB) and Sony Computer Science Laboratory - Paris

steels@arti.vub.ac.be

Abstract

The past decade has seen increasingly more sophisticated experiments to orchestrate the self-organisation of symbolic communication systems on autonomous robots. This is partly due to robots becoming more robust and versatile, thanks to enormous advances in the relevant technologies, and richer in sensory-motor behavior, thanks to the flourishing of the behavior-based approach to robotics. Progress is also due to a much deeper understanding of the issues and mechanisms governing the emergence of language. This talk surveys first the history and state of the art in the field. Language games have turned out to be an enormously fruitful paradigm to frame investigations. The self-organisation of lexicons for naming objects or referring to objects with single or multiple categories is now routinely achieved. The interaction between an emergent lexicon and an emergent repertoire of perceptually grounded categories such as colors has now been clearly demonstrated. We understand how and why a compositional language may arise and how and why words may become structured into syntactic patterns and then into hierarchical structures. Despite all this, real progress in the emergence of natural language like grammars on autonomous robots is still forthcoming. The second part of the talk identifies the issues and what it would take to resolve them.

Proc. LangRo'2007
978-972-96895-2-9

Grammar as System of Second-order Cultural Constraints on Action and Perception: intrinsic functional constraints on language as system of action and representation

Paul J. Thibault

University of Agder, Kristiansand, Norway

pauljthibault@gmail.com

Extended Abstract

Utterances are always context-dependent. This is often taken to mean that utterances stand in some kind of referential or encoded correspondence relationship to whatever it is they refer to in the world 'out there'. Such a view, which typically takes the sentence as its upper level unit of analysis, fails to account for grammar as a system of intrinsic functional constraints on language as a form of social action that is embedded in what Goffman called the 'interaction order'. I take the view, along with Stephen Cowley, that 'grammar' is a virtual system of second-order cultural patterns and digitalized semantic categories. When coupled to the micro-temporal bodily dynamics of first-order languaging in the 'real time' of face-to-face interactive encounters, grammar has the power to guide and modulate the behavioural dynamics of first-order languaging. But it does more than this. It does this job of constraining in ways that shape not only our perceptions and understandings of the behavioural event qua bodily activity (e.g. a vocalization), but also how the action-perception potential of the vocalization links interactants to each other and to their shared worlds in ways that can cognitively transform the agents' relationships to each other and to their worlds. Moreover, this happens in ways that are motivated by the seeking of values on the part of the participants in interaction (Thibault 2004, Hodges 2007). Grammar is therefore normative and suffused with cultural values. It is in many respects a conventionalized system of constraints and patterned regularities, but it is not a fixed and static system which serves to encode meanings (semantics) in grammatical forms. Instead, it is better to think in terms of grammaticalization as emergent and dynamical processes of constraints on diverse time scales in ways partly suggested by Hopper's construction grammar (1987).

The intrinsic functional linguistic organization of utterances and the constraints that such organization entails mean that languaging activity is constrained and guided by the intrinsic constraints on language and other action systems rather than exclusively by constraints that are extrinsic to its mode of organization. Intrinsic constraints are constraints that are intrinsic to the ontology of the system in question and the given system's modes of organization. The ontology of language is that of an action system that is able to act upon and transform both social situations and the cognitive dynamics of the individuals who participate in these situations. Its inherently recursive character furthermore means that language qua conventionalized action system also has the power to act upon and transform itself qua metalanguage.

Grammatical structures are functionally organized structures of action which indicate by virtue of their intrinsic organization how underlying relational networks of representations are to be acted upon and transformed in the course of dialogically grounded and coordinated interaction. Grammatical structures also function to differentiate or partition the world in semiotically and cognitively salient ways on the basis of systems of digitalized semantic categories and systems of classification. To develop this thesis, I shall take one important contemporary functional theory of (second-order) language, viz. systemic-functional linguistics, as pioneered by Michael Halliday (e.g. 1979, 1994 [1985], 2004), and show how some of its core theoretical constructs can be reconstituted in ways that are helpful for explaining grammar as a distributed system of second-order cultural patterns and conventions that constrain action and representation by virtue of grammar's own intrinsically functional organization. I will focus in particular on two functional domains in the grammar of the clause that are postulated by systemic-functional grammar, i.e. the ideational function of transitivity structure (argument structure or case grammar in some other traditions) and the interpersonal-interactive function of mood (declarative, interrogative imperative, etc). Systemic-functional grammar is one of a family of current functional theories – e.g. the west coast functionalism of Thompson and Mann and Dik's functionalism – that see grammar as motivated by and interfacing with discourse or activity levels of organization that go beyond the sentence. Indeed, I will suggest the need to ground grammar in dialogical units rather than in the formalism of abstract sentence-types. This perspective also provides a framework for showing how grammar is embedded in and functional in the large-scale activity structures and genres of a given interaction order.

The grammatical structures of linguistic propositions and proposals qua modes of action (cf. discourse moves) are not encodings of representations of something external to them to which they correspond. Instead, they are modes of action for operating on representations and for transforming them. The Subject-Predicate mode of organization of the mood structure of the clause is in fact a functionally constrained mode of organization to serve that very end, i.e. for acting on and transforming underlying networks of representations and the ways in which interactants orient to and evaluate these. Representations are functional networks of terms (nodes) and the connections between these. Connections between terms construct functional relationships of representation that have the form of a network topology.

Linguistic utterances are operations on these representations and their associated values and value stances. They can add to or modify the representations or provide indications to others as to how to modify them. The intrinsic functional organization of utterances qua forms of action or operations (not representations) is, however, necessary for indicating how relational networks of underlying representations can be acted on and modified. We will see in my presentation how the ideational and interpersonal dimensions of the grammar's intrinsic functional organization work together as a set of grammaticalized resources that enable agents-in-interaction to operate on representations and to coordinate both the actions and the viewpoints of the diverse selves-in-interaction.

The solution I will put forward is that locations within this topological relational network of representations can be specified by differentiating or partitioning them with the overall topology by means of the digitalized semantic categories of natural languages. The transitivity or ideational categories of clause grammar – e.g. transitivity selections in the clause such as, for instance, the semantic constructions [Actor-Process: Action-Goal] or [Agent-Process: Event-Affected] - are differentiations in this sense. A given selection is a semantic differentiator that zeroes in on some aspect of the overall network organization as the current focus of attention or the current concern of the activity. The connections that are made between nodes in the network of representations are therefore activated in ways that correspond to a currently active state of the topology in the form of what Lemke (1983, 1985) and Thibault (1986, 1989) have called networks of thematic relations that are activated and instantiated in discourse. The connections between nodes are functional relationships of particular representations. Grammar plays a pivotal role in organizing these relations, which are primarily grammatical rather than lexical per se. Networks of representations are not simply stored in the heads of individuals, but are distributed and external representational resources; they are cultural affordances that represent and mediate opportunities for acting and perceiving in ways that extend the cognitive and interactive reach of individuals and cultural groups.

P.S. I have no expertise in or knowledge of robotics. However, I will seek in my lecture to show how the kinds of functional constraints intrinsic to grammar as action system should have things to say that are of interest to and of relevance to those interested in designing robots that can participate in languaging activities with humans.

References

Halliday, M. A. K. 1979. 'Modes of meaning and modes of expression: types of grammatical structure and their determination by different semantic functions'. In D. J. Allerton, Edward Carney and David Holdcroft (eds.), *Function and Context in Linguistic Analysis: A Festschrift for William Haas*, 57-79. Cambridge: Cambridge University Press.

Halliday, M. A. K. 1994 [1985]. *Introduction to Functional Grammar*. 2nd edition. London and Melbourne: Arnold.

Halliday, M. A. K. and Matthiessen, Christian 2004. *Introduction to Functional Grammar*. 3rd edition. London and New York: Arnold.

Hodges, Bert H. 2007. 'Good prospects: ecological and social perspectives on conforming, creating, and caring in conversation'. *Language Sciences* 29: 584-604.

Hopper, Paul. 1987. 'Emergent grammar'. *Berkeley Linguistics Society* 13 (1987): 139-157.

Lemke, Jay L. 1983. 'Thematic analysis: systems, structures, and strategies'. *Recherches Sémiotiques / Semiotic Inquiry* (RSSI) 3(2): 159-187.

Lemke, Jay L. 1985. 'Ideology, intertextuality, and the notion of register.' In James D. Benson and William S. Greaves (eds.), *Systemic Perspectives on Discourse*, Volume I, 275-94. Norwood, NJ: Ablex.

Thibault, Paul J. 1986. 'Thematic system analysis and the construction of knowledge and belief in discourse: the headlines in two Italian newspaper texts', In *Text, Discourse, and Context: A social semiotic perspective*, 44-91. Victoria University, Toronto: Monographs, Working Papers and Prepublications of the Toronto Semiotic Circle, Vol. 3.

Thibault, Paul J. 1989. 'Semantic variation, social heteroglossia and intertextuality: thematic and axiological meanings in spoken discourse'. *Critical Studies* 1(2): 181-209.

Thibault, Paul J. 2004. *Agency and Consciousness in Discourse: Self-other dynamics as complex system*. London and New York: Continuum.

Allow me to pick your brain: how language and thought can be shared between robots

Tony Belpaeme

University of Plymouth, UK

tony.belpaeme@plymouth.ac.uk

Language and thought are mutually supportive

In recent years, through experimental studies, it has become clear that language has a definite influence on cognition, and specifically on the use and interpretation of conceptual knowledge (cf. Davidoff, Davies and Roberson 1999; Roberson, Davidoff, Davies and Shapiro 2005; Gilbert, Regier, Kay and Ivry 2006). Many, including me, support a view where language not only has an impact on cognition, but where thought and language are mutually supportive.

The focus of the talk will be the interplay between language and category acquisition. We start from the premise that intelligent systems will need representations which resonate with human representations in order for human-machine interaction to succeed (Steels and Belpaeme 2005; Belpaeme and Cowley 2007). Recent psychological evidence has demonstrated that thought and language are mutually supportive and impact on each other. Category acquisition is facilitated by linguistic interaction, both in infants (Xu 2002; O'Hanlon and Roberson 2007) and adults (Lupyan 2006). This resonates with memetics in the sense that knowledge, present in categories and concepts, is propagated through culture. We believe that cognitive systems should be sensitive to culture and language (with language being the prime medium for memetic propagation) in order to acquire human-like semantic representations.

We will argue that language is not contained within the individual, but is the amalgamated result of producing and interpreting communicative expressions in a population of language users. No single individual has access to the entire language, but has its own interpretation of the population's language. By implication, if we accept that language has an influence on concept formation, concepts are subject to the same constraints: each individual holds an interpretation of a concept. These concepts are private, but are similar enough to concepts of other individuals to allow communication; concepts of individuals are *coordinated* (Steels and Belpaeme 2005). There are several mechanisms through which concepts can become coordinated. Some concepts might be native and as such do not need to be acquired and coordinated. Others, such as perceptual categories, can be acquired by the individual through interacting with the environment. However, acquiring concepts through interaction with the environment does not guarantee the concepts to be sufficiently coordinated to allow communication (for an example on colour see Belpaeme and Bleys 2004). A third mechanism is cultural acquisition of concepts: here concepts are acquired by interaction with one's peers, with language being the most important medium of transmitting conceptual knowledge. It is obvious why this is necessary for abstract concepts, such as DEMOCRACY, but it has been argued that language is also crucial in acquiring perceptual concepts, such as RED, EMPTY or ANGRY. Among others, language helps a developing individual to indirectly access concepts of others — obviously, one does not have direct access to others' concepts. It also allows the learner to access different hypothesis maintained by a range of individuals. This aids learning, and specifically generalisation during learning (cf. machine learning techniques where a large training set supports learning). Finally, language also serves to delimit or constrain conceptual representations, and during concept development linguistic labels act as anchors that prevent concepts from drifting in a semantic sea (cf. Belpaeme and Bleys 2006).

Robots acquiring human concepts

If robots are to interact, communicate and reason with us, they will need conceptual representations which resonate with ours. A straightforward approach is to specify the semantics of internal representation in a machine, but as robots are expected to accrue large amounts of information it will be unlikely that conceptual information can provided through pre-programming semantics.

It has been suggested that a developmental approach could provide a successful approach to achieving artificial intelligence (e.g. Weng, McClelland, Pentland, Sporns, Stockman, Sur and Thelen 2001), and we would like to argue that a developmental trajectory will be needed for robots to acquire grounding language.

When building a robot that acquires the meaning of words, several elements can be shortcut. We are for example not interested in auditory perception or vocal production; instead we will focus on elements that are part of a semiotic schema: perception of external stimuli, mental representations, and the association between mental representations and linguistic labels and structures.

A first requirement is an understanding of learning mechanisms involved. Infants employ a number of biases, constraints or preconceptions which aid in language and meaning acquisition. Several of these have already been uncovered in developmental psychology and linguistics, such as mutual exclusivity constraint (Markman 1989) or novel name-nameless category

978-972-96895-2-9

(N3C) constraint (Mervis and Bertrand 1994). Some constraints require a deeper understanding to allow an implementation on a robot. Others are not understood well enough: for example, the amount of feedback given by the teacher to the learner. The balance between receiving no feedback, receiving implicit feedback or receiving explicit feedback is still not well understood and will be crucial in constructing artificial systems which acquire semantic through human-robot interaction.

Telepathic robots

While humans are not able to access each others' brains to extract knowledge and have to rely on communication to do so, robots do not have this limitation. A robot could easily request information from others robots through channels other than gestural or linguistic channels. If a robot does not know what "dog" is, it could request a human to explain what dog is or it could request for it to be shown a dog. However, it might be more effective to request the meaning of dog from other robots connected through the internet. Such "telepathic" concept acquisition poses a number of opportunities and challenges.

Concept acquisition could be sped up enormously: while children take years to crystallise certain concepts (for example, colour categories are only a mastered around the age of 3 or 4), a robot could download the meaning of certain words from another robot or from several robots, unconstrained by the noisy media of linguistic communication and perception. A robot which did not know the meaning of "dog" could, without ever being shown a dog, learn what a dog is from another robot that has sensory access to dogs.

However, this poses a number of challenges for the learning algorithms. They should be geared towards acquiring and adapting semantic representations according to linguistic interaction with humans, but at the same time should allow semantic information to be injected from other robots. Another issue concerns embodiment: the conceptual representations of a robot will be tightly coupled to its embodiment, i.e. its sensors and actuators. However, how does one integrate concepts from a robot with a different embodiment? Embodiment can vary radically, such a robot having a different type of camera or a different mode of moving through its environment, but even robots which are identical will have slight variations in embodiment due to noise on sensors and actuators which might have far reaching implications for its semantic concepts.

We wish to study these and related issues in a number of project, most notably the ITALK project sponsored by Europe under the 7th framework programme; of which a brief overview will be given.

References

Belpaeme, T. and Cowley, S. J. 2007. Extended Symbol Grounding. *Interaction studies: Social behaviour and communication in biological and artificial systems*, **8**, (1) 1-6.

O'Hanlon, C. G. and Roberson, D. 2007. What constrains children's learning of novel shape terms? *Journal of Experimental Child Psychology*, **97**, 138-148.

Gilbert, A., Regier, T., Kay, P. and Ivry, R. 2006. Whorf hypothesis is supported in the right visual field but not the left. *Proceedings of the National Academy of Sciences*, **103**, (2) 489-494.

Lupyan, G. 2006. Labels facilitate learning of novel categories. In: Cangelosi, A., Smith, A.D.M. and Smith, K. (Eds). *The Evolution of Language.* World Scientific. pp 190-197.

Belpaeme, T. and Bleys, J. 2006. The subtle interplay between language and category acquisition and how it explains the universality of colour categories. In: Cangelosi, A., Smith, A.D.M. and Smith, K. (Eds). *The Evolution of Language.* World Scientific. pp 395-396.

Roberson, D., Davidoff, J., Davies, I. R. L. and Shapiro, L. R. 2005. Color categories: Evidence for the cultural relativity hypothesis. *Cognitive Psychology*, **50**, (4) 378-411.

Steels, L. and Belpaeme, T. 2005. Coordinating Perceptually Grounded Categories through Language. A Case Study for Colour. *Behavioral and Brain Sciences*, **24**, (8) 469-529.

Belpaeme, T. and Bleys, J. 2004. Does structure in the environment influence our conceptualization? Proceedings of the Evolution of Language Conference 2004, Leipzig, Germany.

Xu, F. 2002. The role of language in acquiring object kind concepts in infancy. *Cognition*, **85**, 223-250.

Weng, J. Y., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M. and Thelen, E. 2001. Artificial intelligence - Autonomous mental development by robots and animals. *Science*, **291**, (5504) 599-600.

Davidoff, J., Davies, I. and Roberson, D. 1999. Colour categories in a stone-age tribe. *Nature*, **398**, 203-204.

Mervis, C. B. and Bertrand, J. 1994. Acquisition of the Novel Name--Nameless Category (N3C) Principle. *Child Development*, **65**, (6) 1646-1662.

Markman, E. M. 1989. *Categorization and naming in children: problems of induction.* The MIT Press, Cambridge, MA.

How words become cognitive

Rob Clowes[1], Carlos Herrera [2], Martin McGinnity[2], Tom Ziemke[3]

[1]COGS, University of Sussex
[2]Intelligent Systems Research Centre, U. of Ulster
[3]University of Skövde, School of Humanities and Informatics, Sweden
robertc@sussex.ac.uk

Abstract

This paper explores the thesis embraced by authors such as Vygotksy and Clark that considers public language to be an external cognitive resource that may be internalised. We conceive such a process as involving the immersion of a developing agent in a language mediated leaning task. The internalisation of language can thus be explained in the context of substituting external instructions with auto-generated ones for self-guidance. The experiments reported aim to help build an understanding of the nature of this process within the framework of an embodied and embedded approach to mind.

1. Ways of understanding the cognitive conception of language

Vygotsky argued that language was internalised and thus came to play a pivotal role in constructing mind. While this idea lay dormant in cognitive science for a long time, it has recently started to be treated much more seriously both theoretically (Clowes, 2007a, 2007b), and also by robotic modellers (Lindblom & Ziemke, 2003; Mirolli & Parisi, in press) as a way of explaining the emergence of complex cognitive architectures.

There is some persuasive evidence that language does indeed play some cognitive role i.e. (Hermer-Vazquez, Spelke, & Katnelson, 1999). In Hermer-Vasquez et al's study children are placed in a room where they attempt to find treats. The task involves the children needing to combine geometric and colour information, which is quite a hard task for them. In Hermer-Vasquez et al's study talking-to-oneself appears to be the only indicator of whether children are able to solve the task is they talk to themselves about it. In many ways this task seems to confirm Vygotsky's theory that self-directed, though ultimately sub-vocal, speech can play a central role both synchronically and diachronically (developmentally) in higher cognition. While many theorists now agree that there is some cognitive role for natural language there is considerably less agreement on either how it works or how the role is taken up.

There are some interesting debates between (Carruthers, 2002) and (Clark, 2006b) about how language comes to play its cognitive role, especially around whether it might be best conceived of as facilitating or transforming our native or pre-linguistic information processing architecture or mainly seen as playing a role in extending and re-using that architecture in a novel way.

According to Carruthers (2002) the role of language is to translate information between otherwise encapsulated modules. Carruthers argues for a view he calls 'Central-Process Modularism' or CPM. Conceived against the background of the idea of massively modular minds and evolutionary psychology CPM is the idea that the mind is modular not only in its input systems[1] but also in its central conceptual capabilities; the mind is a Swiss-army knife all the way up. On this view each central conceptual systems has its own proprietary representational code and, at least for the main conceptual modules, these rely on propositional encodings. For Carruthers the cognitive role of language can be regarded as a sort of lingua franca for communication amongst modules. Modal quasi-thoughts come to be integrated by the use of the language module recruited to a new cognitive role.

Andy Clark offers a different view on how to understand the role of language. For him the brain is a connectionist pattern completion device embedded in a structuring body and world. Cognitive capabilities are probably not strongly modularised, and cognition in any case extends beyond the body and brain (Clark & Chalmers, 1998). Much cognitive activity, needs to be understood as tool-using and world-involving (Clark, 2003). Amongst the most important of tools is language and the roles it plays are multiform (Clark, 1998). Playing these roles hinges upon the way that language can provide a better 'material' environment for thinking in (Clark, 2006a), as much as providing an 'internal' environment to think with.

The basic problem with the Clark approach is that it becomes difficult to conceptualise how language makes the move beyond being an external tool to an 'internal'[2] self-regulatory device. Indeed as is foregrounded in Clark's debate with Wheeler (cf. Clark, 2004; Wheeler,

[1] As argued for instance by (Fodor, 1983).

[2] The very word internal here can be somewhat tricky. In the Vygotskian tradition there are two distinct phases of internalisation. The first, where language takes on a psychological role for the child in virtue of being used in ego-centric speech to play a role in the self-control of the child, and the second where language becomes interiorized and fused with thought. For some further analysis of this notion see (Wertsch & Addison Stone, 1985)

2004) it becomes something of a problem as to what it might mean to internalise language on this account. In part this paper will attempt to bridge this gap by a detailed a discussion of some robotic experiments that seek to explore how the process of internalisation might get underway.

The robotic simulation experiments described below constitute some first steps towards investigating this issue computationally. Using these minimal simulations, we hope to gain a better conceptual grasp on how it might be possible for language to take up a cognitive role in mind if our brains start out as the sorts of embodied and distributed pattern completers we find in much contemporary theorizing.

2 Language as cognitive niche

2.1 Learning an inherited task

Vygotsky understood the exposure of the child to existing linguistic forms of communication as mediators of action and mediators that can ultimately be appropriated to self-control. As these regimes are internalised they can come to be recalled without any external linguistic input. Clark summarises Vygotsky observations as follows:

"When the child, confronted by a tricky challenge, is `talked through' the problem by a more experienced agent, the child can often succeed at tasks which would otherwise prove impossible (think of learning to tie your shoelaces). Later on, when the adult is absent, the child can conduct a similar dialogue, but this time with herself. But even in this latter case, it is argued, the speech (be it vocal or `internalized') functions so as to guide behavior, to focus attention, and to guard against common errors. In such cases, the role of language is to guide and shape our own behavior -- it is a tool for structuring and controlling action and not merely a medium of information transfer between agents." (Clark, 1998)

This role of 'internalised' language emerges as a logical consequence of the properties of public language in structuring behaviour. In other words, if public language is an external scaffolding for structuring behaviour, internal language may play the same role. The idea is not far from the classic cognitive science approach of explaining intelligence as the product of building models of the world. But the main difference is that now the mental models that allow us to imagine or anticipate external resources are rooted in a externally aided and language mediated learning process. "We can be instructed to behave in a particular way. Responding to instructions in this way can be viewed simply as responding to some environmental event. We can also remember such an instruction and tell ourselves what to do. We have, in this way, internalized the instruction. We believe that the process of following instructions is essentially the same whether we have told ourselves or have been told what to do. Thus even here we have a kind of internalization of an external representational format."
(Rumelhart, Smolensky, McClelland, & Hinton, 1986 p. 43)

The focus of this approach is therefore not in revealing the inner architecture of the biological mind, but to understand an adaptive process that involves a social structure that remains across generations. Several aspects of this process can be stressed across different time-scales:

- This is a process of ontogenetic learning, or more properly epigenesis, in which the embodied agent's ability to learn and develop involves the deep interaction of biologically specified tendencies, and environmental structures and scaffolding.
- At a larger time-scale (across a few generations) this is a process in which each agent inherits a structure of cognitive scaffolding through public language.
- At a phylogenetic level, the species evolves the ability to be integrated in a linguistic niche.

It is these conditions that we attempt to replicate, in order to investigate whether it is plausible to suggest that such learning process would converge on a strategy that allows the agent to replace external guidance by internal language. In order to remove designer bias from the experimental design, we use self-organising techniques, combining evolutionary robotics with learning in neural networks.

One essential aspect of language lies, as Vygotsky suggested, in how it must be learned before it can be mastered. For this reason, the study of the relationship between learning and evolution provides an interesting background for the emergence of linguistic forms, such as internal language. An embodied agent needs to adapt both to persistent features of the environment (through evolution) and changing patterns in its interaction with the environment (through learning). Species can be differentiated as of the degree of learning that individuals go through to shape their adaptive features. The need for learning can be seen as corresponding to the richness and variability of the species niche or *Umwelt* (Uexküll, 1957; Tom Ziemke & Sharkey, 2001). The concept of Umwelt refers to the environment insofar it is cognized by the inhabiting agent. On one side of the scale, we find micro organisms and even insects, flexibly adapted to persistent niches. Their Umwelt is simple and specific, and it is comprehended within basic sensory-motor coordination patterns or functional circles. On the other hand, we find humans, whose Umwelt is not restricted to such fixed patterns. The niche, for a human, does not only consist of environmental features, but includes rich changing social structures, such as language (Clark, 2006a). The need for language can therefore be understood, not only, and perhaps not even primarily for communication, but as the carrier of mechanisms involved in behavioural and cognitive control; a property

Proc. LangRo'2007
978-972-96895-2-9

of the defining niche (cf: Laland, Odling-Smee, & Feldman, 2000; Sinha, 2006).

It is, in part, the variability of language that makes learning an essential feature. As any social structure, language survives in the shared use individuals make of it, and it is "handed" from generation to generation. Inevitably, languages change, die or emerge. Although languages and the capacity to learn them probably co-evolve, cf (Deacon, 1997), every newly born baby is "a blank sheet" at least to the extent that it is ready to learn any language. Being embedded in a linguistic world is therefore tied to having the ability to learn forms of linguistic acts beyond what can be achieved with innate behaviours.

It is therefore possible to assume that an agent learns a language through being embedded in different forms of communication within different social structures. For instance, a series of actions following linguistic input may need to provide some form of replacement (internal language) in order to carry out the same actions. That would imply that, if a robot is to inhabit a human niche, and thus frame its actions in a linguistic environment, it may need to develop the ability to internally replace, simulate or imagine the linguistic environment.

3 – Experimental background

Evolution and learning are also major design techniques for highly distributed artificial systems, and have been widely applied in the fields of neural networks and neuro-robotics. Artificial neural networks are often defined by a network architecture, weighted connections between neurons and neuron activation. A neural network may use different rules, such as the Hebbian rule or back-propagation, to alter the weights between different neurons, depending on the current activation of the network. This method has proven extremely successful in adjusting the network functionality to different patterns of stimulation, and, in the case of robots and embodied networks, interaction with the environment.

Artificial evolution of neural networks, on the other hand, often assumes that weights between neurons are fixed during lifetime, and selects the weights that produce best overall results, given some criteria, through a selection mechanism. Artificial evolution, nevertheless, is not restricted to the selection of weights. It could be applied to the selection of any relevant parameter, for instance the network architecture, embodiment in the case of robots, or learning rules in the case of plastic neurocontrollers.

It would be a mistake to conflate artificial evolution and learning with the phenomena as present in biological systems. Yamauchi and Beer (1994), for example, have shown how evolved non-plastic controllers can appear to perform reinforcement learning. Neural plasticity, on the other hand, can be used to shape the overall features of an agent, based on patterns of sensory-motor coordination with no fitness evaluation or genetic mutation (as in e.g. Morse, submitted). Even though some important features are present, the natural phenomena are still vaguely understood. One of the differences is that, in natural agents, evolution and learning are not two competing adaptive mechanisms, but necessarily co-occurring phenomena. The distinction between mechanisms responsible for generating behaviour and those responsible for learning "… is difficult to defend biologically, because many of the same biochemical processes are in involved in both processes" (Yamauchi and Beer 1994, p.243).

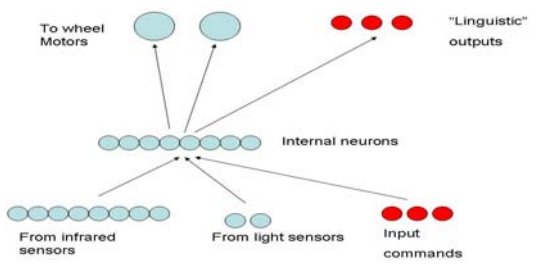
The relationship between evolution and learning in neuro-robotics has been investigated by several researchers. For instance, (Nolfi, Elman, & Parisi, 1994) demonstrates how a population selected on one task may increase their performance when, at an individual level, an individual learns a task differently to the one it is selected for. The experimental setup in this paper follows up the one described in Floreano and Mondada 1996. Rather than evolving the weights of a neural network, the authors evolve a genotype that encodes how the network (with a fixed architecture) should modify its weights during lifetime. Concretely, they allow evolution to mutate the learning properties of each individual synapse, as what type of Hebbian learning rule it uses and what is the learning rate.

3.1 Robot and architecture

The following experiments were carried out in a modified version of Evorobot, developed by Stefano Nolfi. Evorobot is a Khepera simulator that incorporates a genetic algorithm and neural networks.

The agent's controller has the following structure. The network receives sensory inputs from infrared (8 sensors) and light (front and rear) sensors. The network has a number of internal neurons (8 or 12), and two motor outputs (left and right motors). The network also receives three extra inputs (intended to represent linguistic instructions from an external agent, in this case the experimenter), and produces three extra outputs, which can be used to replace the external instructions. The linguistic inputs can take the values of either 0 or 1.

Figure 1 - Neural Architecture



The method employed in the following experiments replicates the one reported in (Floreano and Mondada 1996), where the authors investigate how neural mechanisms underlying

ontogenetic learning are themselves developed and shaped by the evolutionary process.

A simple genetic algorithm is used to generate new genotypes. At the beginning of each epoch a chromosome is decoded into the corresponding neural controller, and its performance evaluated. Each of the genotypes contains enough information to generate a controller, given the architecture represented in fig 1. Where in many experiments this is achieved by encoding in the chromosome the weight that defines each synapse, this method the genotype encodes the neural architecture and learning rules of each of the 144 synapses.

The four allowed learning rules were: pure Hebbian, Postsynaptic, Presynaptic and Covariance (based on Willshaw & Dayan, 1990). The learning rate could take four different values {0.0, 0.3, 0.7, 1.0}. The two other properties are whether each synapse is excitatory or inhibitory and whether it drives or modulates the postsynaptic neuron., and the weights of the network synapses are initialised to small random values. A chromosome would therefore contain 6 bits per synapse (2 for rule, 2 for rate, and 2 more for the other 2 properties)

Given the initial random weight, each synapse changes its weight according to the conditions specified in the chromosome (with the exception that weights are constrained to a maximum of 1). More details on the method can be found in (Floreano & Mondada, 1996)

3.2 Environment and task

The environment consists of a plus maze; four corridors that converge to a central area illuminated by a light bulb, as represented in the Fig 2. In the simulated environment, the end of the corridors are blocked with obstacles. The robot is, at the beginning of its lifetime, placed at the end of the west corridors, and the task it must achieve is to navigate the corridor towards the central area, and then turn to the left towards the end of the north corridor. Once this is achieved, the robot is "transported" to the end of the south corridor, where it needs to achieve the same task, go to the centre, but now turn right to arrive at the end of the east corridor.

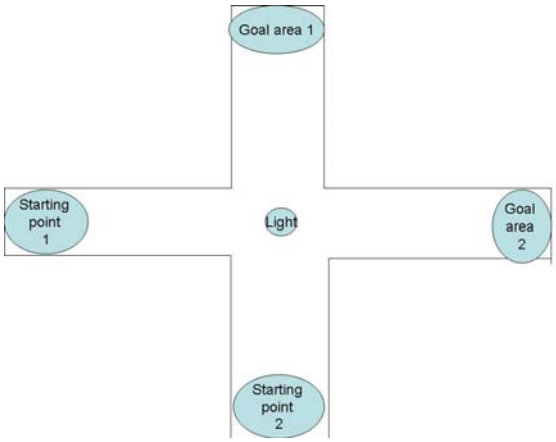


Figure 2 - The Environment

During the training phase, individuals are rewarded for their ability to perform both tasks (go north from west, the east from south). During this phase, in addition to the inputs from the sensors embodied in the robot, the robot receives three signals (either 0 of 1 for three extra input nodes), which are intended to represent three commands (go_to_light, turn_left, turn_right). As received for the robot, they are simply three signals with no significance. The ascription of meaning rests on the potential grounding for the meaning of these symbols in the action repertoire of the agent. We expect evolution to find a grounding for such symbols insofar as instructions will be given when required in skilled interaction.

We call this phase *the training phase* because during its course the network weights are updated following some evolved learning rules and rates, given some initial random weights. If the robot has completed the learning phase, it enters the trial phase, where weights are "frozen" to their values after the learning phase, i.e. they are not allowed to change. Instructions cease, and as an alternative the state of the internal outputs (normalised to 0 or 1) takes their place in the network structure. We call this *the trial phase*. The tasks are carried out in exactly the same manner as in the trial phase, and extra time is awarded to individuals that complete the trial phase, in order to repeat the task.

The fitness function is given in the following table

	Learning phase	Trial phase
Go north from west	1	5
Go east from south	2	6

The rationale behind this fitness function is to allow the tracking of what tasks have been achieved, and to encourage agents to perform well on the trial phase. After several hundred generations, evolution consistently found genotypes that allowed for the learning task to be completed, as well as the first task of the trial phase, scoring a total of 8. In several occasions we found robots that completed the task several times, scoring up to one hundred points, but these were not robust (they did not allow replication, possibly because they could not overcome the randomisation of initial weights). Nevertheless, the experimental results show some interesting points that can inform current debates about how language takes up its cognitive role. In what follows we show graphical depictions of the robot in several trials as these help illuminate one possibility for this discussion.

Proc. LangRo'2007
978-972-96895-2-9

4. Results

4.1 Learning phase: grounding instructions in embodied interaction

Here a simulated Khepera begins the cycle of its behaviour where it has external commands fed to it. The Khepera is moving from left to right toward the bulb at the centre of the arena and it receives an externally generated (to the agent) signal in the input labelled *centre*. This signal is given in the learning phase whenever we intend the robot to navigate to the light. As the Khepera encounters the object in the centre of the maze, it receives a new instruction represented in *left* that can be glossed as turn left. The Khepera responds to this instruction (and the other data about the situation in which it is embedded) and turns left (see Fig 3).

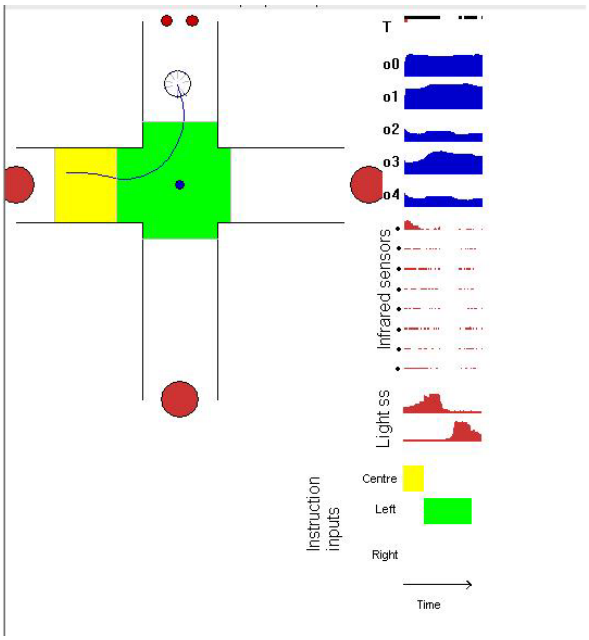


Figure 3. Agent begins movement through maze, receiving instruction 'go_to_centre'. Once in the central area veers left in response to the command 'turn left'.

Next (Fig 4) the Khepera has been transported to the bottom end of the maze, gets an input once more in I10, signalling the robot to movement forward toward the top of the maze. The robot moves forward and, as it approaches again the centre area, receives an instruction in *right* (the last externally generated signal it received), and it successfully responds to this by turning to the right.

As explained before, any Khepera that has successively managed to run the maze now receives a total of 3 points and enters the trial phase.

4.2 Trial phase: Re-using commands to scaffold one's own behaviour

As we mentioned above, the main differences between the learning phase and the trial phase are that during the latter weights are fixed to the last value of the training phase and that instructions are not given thereafter, instead replaced by an internal stream. During the learning phase, the weights of the neural network (initialised to random numbers between 0 and 1), are changed given the learning type and rate. Once the agent has performed the task once

(i.e it has gone to the light, if the instruction was 0, to the area if the instruction was 1), there is no more learning or instructions. The weights are fixed and the instructions are replaced by the internal output, or self-directed "language for thought".

In the next diagram (fig 5) we can note that the robot navigates to the centre of the maze without producing the signal go_to_light. It completes the whole task producing a self-directed signal that replaces the external command turn_left, that allows it to turn left at the appropriate moment.

Unfortunately, controllers could not consistently solve the go_east task. The robots continue delivering the same instruction (go_left) at the next stage of its task and merely crashes into the wall to its left. At least on this run the Khepera is unfortunately only able to keep producing the turn left instruction and does indeed

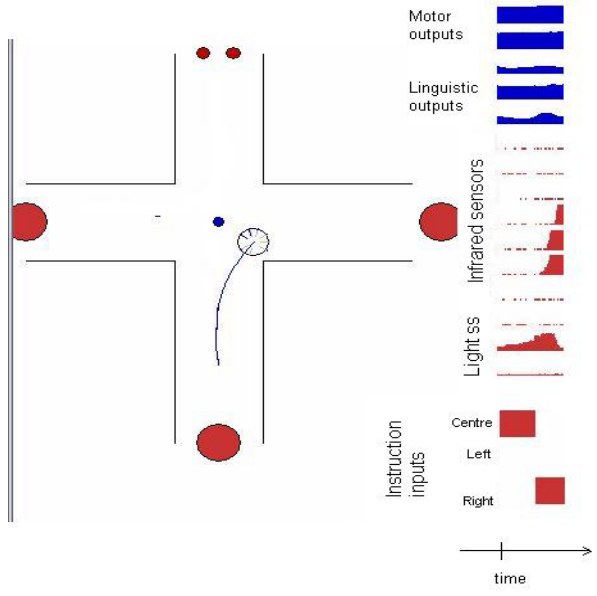


Figure 4. The robot receives the 'veer right' signal and proceeds to do so.

"obey" this self-directed instruction and crash into the wall.[3]

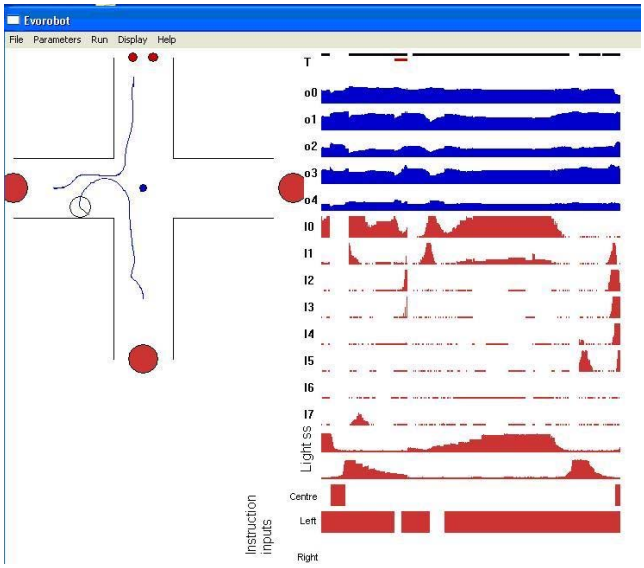


Figure 5. Trial phase

The thing which is interesting about this work is that it opens the possibility that systems which have developed the capability to respond to commands come to, rather easily, pre-produce virtual commands that help them tackle refined versions of the problem. This would be similar to the experiments of Ziemke, Jirenhed and Hesslow (2005) in which simulated robots were evolved to navigate 'blindly' in a simple environment, i.e. based on internally generated rather external sensory input. So on this account, there is not much difference between the capacity to re-use fragments of language to re-structure behaviour, and the capacity to bring to mind other environmental encounters that are of use

Once the capacity to respond to commands is developed it seems only a small step to re-using them in other contexts, at least in a rather basic way[4]. It is at least plausible that words once internalised into ongoing cognitive activity in this way, could begin to play increasingly sophisticated roles in mediating many new forms of online cognitive activity.

Provisional Conclusions and Future Work

Cognitive and evolutionary robotics has for some time attempted to understand the way that an agent adapts to a cognitive niche. It has recently begun to investigate ways that an agent might adapt its ecological niche as a route to self-control (Tom Ziemke, Bergfeldt, Buason, Susi, & Svensson, 2004). Here we have attempted to investigate another departure in this line of thinking, i.e. to understand how simplified language-like activity can be appropriated from the cognitive niche for self control; or to put this in a slightly more Vygotskian language, the internalisation of the social means of self-control.

We demonstrate how in a minimal robotic environment it is possible to model the way that language might move from an inter-cognitive to an intra-cognitive role. An idea that was central to Vygotsky's understanding of the role of language in thinking (Vygotsky, [1934] 1986). This work can also be understood as showing how simple languages can play a role in cognition even in the absence of any supposed internal languages of thought (Fodor, 1975) or indeed the sort of modularist languages of thought proposed by Peter Carruthers (Carruthers, 2002). Perhaps, as Andy Clark writes

> "recalled words and sentences […] act […] less like inner data structures, replete with slots and apt for combinatoric action, and more like cheap ways of adding task-simplifying structure to the perceptual scene. Words and sentences, in this view, act as stable anchor points around which complex neural dynamics swirl and coalesce." (Clark, 2002) (p25)

Within the context of the current theoretical debate (along with work already presented in Clowes & Morse, 2005) this work offers something new. It is as an existence proof of something very interesting, namely the possibility that language can play a role in the building of and sequencing of cognition and of transferring external scaffolding into internal use. While they certainly cannot be taken as deciding this question conclusively one way or another, they can be taken as *opaque thought experiments* (Di Paolo, Noble, & Bullock, 2002), i.e. a means of rendering more apparent the implication and possibilities of certain conceptual models; in this case how language can play a role in thought without assuming anything like the classical view of cognitive architecture.

These minimal simulations start to indicate one way that these words can start to act in a cognitive role. Moerover they indicate how this process can get started without a commitment to a pre-existent propositional Language of Thought (even the modular version favoured by Carruthers), or nativism about language. This is of course not proof that humans do not do things very differently, nor does it demonstrate by itself how

[3] This is not to say however that it could not in a different learning regime manage this. In fact, we have observed that changes in number of sensory inputs or internal neurons can have local effects on the evolutionary space. Given the conditions of the experiment, we found that 8 or 12 internal neurons were "good numbers", and that networks with 9 or 11 internal neurons would hardly evolve individuals capable of solving even the first task.

[4] This raises the question of why we do not find dogs and other pets and domesticated animals re-using commands to self-trigger toward a useful behaviour. It is of course possible that they do, remembering the command of their owner to "stay off the table" and do in fact bring this to mind when the master is no longer there. Such entry-level internalisation is then perhaps open to many trained and social animals

Proc. LangRo'2007
978-972-96895-2-9

language would take thought much beyond the here-and-now of direct cognitive engagement. But it does help develop an alternative way of understanding the way language enters cognition and perhaps provides a bridge to understanding the more elaborate forms of language-involving thought to which human beings seem to owe so much. That is, if you are not already committed to the strong nativist assumptions about propositional languages of thought and indeed the cognitive role of language itself.

References

Carruthers, P. (2002). The cognitive function of language. *Behavioral and Brain Sciences, 25*(06), 657-674.

Clark, A. (1998). Magic Words: How Language Augments Human Computation. In P. Carruthers & J. Boucher (Eds.), *Language and Thought. Interdisciplary Themes* (pp. 162 - 183). Oxford: Oxford University Press.

Clark, A. (2002). Anchors not inner codes, coordination not translation. *Behavioral and Brain Sciences, 25*.

Clark, A. (2003). *Natural Born Cyborgs: Minds, Technologies and the Future of Human Intelligence.* New York: Oxford University Press.

Clark, A. (2004). Is language special? Some remarks on control, coding, and co-ordination. *Language Sciences, 26*(6), 717-726.

Clark, A. (2006a). Language, embodiment, and the cognitive niche. *Trends in Cognitive Sciences, 10*(8), 370-374.

Clark, A. (2006b). Material Symbols. *Philosophical Psychology, 19*(3), 291-307.

Clark, A., & Chalmers, D. (1998). The Extended Mind. *Analysis, 58*, 10-23.

Clowes, R. W. (2007a). A Self-Regulation Model of Inner Speech and its Role in the Organisation of Human Conscious Experience. *Journal of Consciousness Studies, 14*(7), 59-71.

Clowes, R. W. (2007b). Semiotic symbols and the missing theory of thinking. *Interaction Studies, 8*(1), 105-124.

Clowes, R. W., & Morse, A. (2005). Scaffolding Cognition with Words. In L. Berthouze, F. Kaplan, H. Kozima, Y. Yano, J. Konczak, G. Metta, J. Nadel, G. Sandini, G. Stojanov & C. Balkenius (Eds.), *Proceedings of the 5th International Workshop on Epigenetic Robotics* (pp. 102-105). Nara, Japan: Lund University Cognitive Studies, 123. Lund: LUCS.

Deacon, T. W. (1997). *The Symbolic Species: The Co-Evolution of Language and the human brain*: The Penguin Press, Penguin Book Ltd.

Di Paolo, E. A., Noble, J., & Bullock, S. (2002). *Simulation models as opaque thought experiments.* Paper presented at the Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life.

Floreano, D., & Mondada, F. (1996). Evolution of plastic neurocontrollers for situated agents. *From Animals to Animats, 4*, 401-410.

Fodor, J. (1975). *The Language of Thought*. Cambridge, MA: MIT Press.

Fodor, J. (1983). *The Modularity of Mind*. Cambridge, MA: MIT Press.

Hermer-Vazquez, L., Spelke, E., & Katnelson, A. (1999). Sources of flexibilty in human cognition: Dual task studies of space and language. *Cognitive Psychology, 39*, 3-36.

Laland, K. N., Odling-Smee, J., & Feldman, M. W. (2000). Niche construction, biological evolution, and cultural change. *Behavioral and Brain Sciences, 23*, 131-175.

Lindblom, J., & Ziemke, T. (2003). Social Situatedness of Natural and Artificial Intelligence: Vygotsky and Beyond. *Adaptive Behavior, 11*(2), 79-96.

Mirolli, M., & Parisi, D. (in press). Towards a Cognitive Robotics: The Role
of Language as a Cognitive Tool. *New Ideas in Psychology*.

Morse, A. (submitted). Dynamic Liquid Association: Complex Learning Without Implausible Guidance. *The Neural Networks Journal*.

Nolfi, S., Elman, J. L., & Parisi, D. (1994). Learning and Evolution in Neural Networks. Adaptive Behavior, Vol. 3, 1, pp.5-28. *Adaptive Behavior, 3*(1), 5-28.

Rumelhart, D. E., Smolensky, P., McClelland, D., & Hinton, G. E. (1986). Schemata and Sequential Thought Processes in PDP. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition 2* (pp. 7-57): Cambridge MA: MIT Press.

Sinha, C. (2006). Epigenetics, Semiotics, and the Mysteries of the Organism. *Biological Theory, 1*(2), 112-115.

Uexküll, J. (1957). A Stroll through the Worlds of Animals and Men. A Picture Book of Invisible Worlds. *1957): Instinctive Behavior. The Development of a Modern Concept, New York, International Universities Press, Inc*, 5-80.

Vygotsky, L. S. ([1934] 1986). *Thought and Language* (Seventh Printing ed.): MIT Press.

Wertsch, J. V., & Addison Stone, C. (1985). The concept of internalization in Vygotsky's account of the genesis of higher mental functions. In J. V. Wertsch (Ed.), *Culture, communication, and cognition* (pp. (pp. 162-179)). Cambridge, England: Cambridge University Press.

Wheeler, M. (2004). Is language the ultimate artefact? *Language Sciences, 26*(6), 688-710.

Willshaw, D. J., & Dayan, P. (1990). Optimal plasticity in matrix memories: What goes up must come down. *Neural Computation, 2*(85-93).

Yamauchi, B., & Beer, R. (1994). Sequential Behavior and Learning in Evolved Dynamical Neural Networks. *Adaptive Behavior, 2*(3), 219-246.

Ziemke, T., Bergfeldt, N., Buason, G., Susi, T., & Svensson, H. (2004). Evolving Cognitive Scaffolding and Environment Adaptation: A new research direction for evolutionary robotics. *Connection Science, 4*(19), 339-350.

Ziemke, T., Jirenhed, D. A., & Hesslow, G. (2005). Internal Simulation of Perception: A Minimal Neuro-Robotic Model. *Neurocomputing, 28*, 85-104.

Ziemke, T., & Sharkey, N. E. (2001). A stroll through the worlds of robots and animals: Applying Jakob von Uexküll's theory of meaning to adaptive robots and artificial life. *Semiotica,, 134*(1-4), 701-746.

Language and the person problem

Stephen J. Cowley

University of Hertfordshire, UK
and
University of KwaZulu-Natal, South Africa

s.j.cowley@herts.ac.uk

Extended Abstract

Let's ask how world-models give thoughts *objective validity*. Playing down symbols, we can aim to simulate how co-action routines emerge from making use of symbolic *material* (dynamic patterns). Given action-guidance representations (Anderson & Rosenberg, 2007), social robots could mimic children by co-constructing such routines. To sketch the process, I consider how children react when a robot is introduced to their ecology. They link circumstances to extra-somatic markers such as words, nonverbal expression and robot behaviour and, over time, develop congruent ways of co-acting. Linking their inner motivations to the local norms, they develop manifest shared values (e.g. showing interest; teasing).

Given the distributed view of language, robots can be used to set a new agenda for cognitive science. We can model persons as bio-robots who integrate their own motivations with local norms in ways that lead to the rise of co-action. To become persons, we use local values to control our actions. In MacDorman's (2007) terms, cognitive science can pose the *person problem*: "How can robot bodies construct themselves into quasi-persons by attuning to norms in the environment?"

To illustrate how persons develop shared values, I consider a longitudinal study of a social robot in a classroom. The machine changes the local ecology in that, in spite of disappointments, the children learn to value the robot. They develop shared perspectives manifest in co-actional routines. The machine functions as a social mediator (Nabe et al., 2006). It permits children to use full-bodied languaging to enact relationships based on recurrent social routines. Not only does this show why children are receptive to social robots, but it suggests that, if robots are to be truly social, they must (at least) simulate construction of co-action routines. Robots need to build motive systems by mimicking how children use actions to establish norms and values that link their needs with the cultural setting.

Cowley (2007) describes *human symbol grounding* in analogous terms. As part of a distributed cognitive system, the child uses extra-somatic markers in aligning to what others want. By the second year, these markers map onto *words* (e.g. 'more' or 'phu'). Later, the patterns contribute to routines that give rise to adult-like *hearing*. Eventually, the child uses these valued patterns to listen to and, eventually, regulate co-action. As persons, we believe in words: we come to *take a language stance*. This happens as language is grounded into: (a) brains, (b) a changing grasp of practices; and (c) first-person experience. Humans, as MacDorman (2007) suggests, neither are nor embody physical symbol systems. Rather, we gain control over own actions as we self-construct action-guidance representations. Later, we use our experience of hearing to monitor what we both do and say while inhibiting action tendencies. Given (inner) models of interaction-models (sic), we learn to anticipate speaking and, thus, guide our own action. While Vygosky suggests that what goes underground is language, Clowes (2007) shows that higher-level control can arise from the monitoring of real-time action and speech.

Can robots use rewards to interact in ways that sensitise them to the physical words or extra-somatic cues that mark local values? Can they be used –not just in routine interaction but to set up congruent forms of co-action? In pursuing this ambitious goal, we can learn from Rodney Brooks and his collaborators. Just as they put the mind back into the body, bodies can be built to engage with human practices. Robot designers can simulate use of the cultural ecology.

Distributed systems (both robots and multi-agent simulations) can be used to simulate aspects of culturally embedded intelligence. Above all, they can be used to generate and test hypotheses about the rise of congruent co-action.

(1) Multi-agent systems can simulate how congruent and valued co-action derives from integrating symbolic material with routine interaction.
(2) Non-social robots can use perception and action to correlate interactional (and phonetic) patterns with real-world invariances.
(3) Social robots can be designed to interface with humans as action-guided representations serve in tracking how, in some circumstances, co-action is shaped to symbolic constraints.

Instead of positing that mind is machine-like, we can take it that bio-systems interact with world by building interaction models (action guidance representations). However, *persons* also learn to act in accord with

reasons. Given skills in using social norms to monitor and motivate action, we develop models of interaction-models (we *speak* and *imagine* speaking). These skills arise from co-action based in shared experience of valued markers. In Craik's (1943) terms, 'external processes' enable us to come up with 'symbols' (extra-somatic markers) that can be construed in ways that, at times, have 'objective validity'. To pursue this insight, robots can be built to develop control systems as they gain from exploiting distributed language. By using interactional history, they would learn to act more as we do. Like persons, they might even discover the power of the language stance.

References

Journal Articles

Anderson, M. & Rosenberg, G. (to appear). Content and action: The guidance theory of representation. In D. Smith (ed.) *Evolutionary Biology and the Central Problems of Cognitive Science*, *Journal of Mind and Behavior,* 2007.

Clowes, R. (2007). A Self-Regulation Model of Inner Speech and its Role in the Organisation of Human Conscious Experience. Journal of Consciousness Studies, 14/7: 59-71.

Cowley, S.J. (2007). How human infants deal with symbol grounding. *Interaction Studies, 8.1: 83-104.*

MacDorman, K. (2007). Life after the symbol-system metaphor. *Interaction Studies 8.1: 143-158.*

Books

Craik, K. (1943). *The Nature of Explanation*. Cambridge: Cambridge University Press.

Proceedings Paper

Nabe, S., Cowley, S.J., Kanda, T. Ishiguro, H. Iraki, K. & Nargita, N. (2006). Robots and social mediators: coding for engineers. *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication. University of Hertfordshire*, pp. 384-390.

Learning User Models with Limited Reinforcement: An Adaptive Human-Robot Interaction System

Finale Doshi and Nicholas Roy
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology,
32 Vassar St., Cambridge, MA 02139.
{finale|nickroy}@mit.edu

Abstract

Spoken dialog allows for natural human-robot interaction, but ambiguous phrases or noisy speech recognition can lead to considerable uncertainty during the conversation. Planning algorithms such as the Partially Observable Markov Decision Process (POMDP) have successfully overcome this uncertainty and generated reasonable interactions during natural dialogs between people and mobile robots. However, like all dialog systems, a POMDP is defined by a large number of parameters that may be difficult to specify a priori from domain knowledge. Even with an online adaptive system, learning these parameters may require a tedious training period from the user.

In this paper, we present an approach which lets the agent decide when it needs more information to be an effective dialog manager. If the agent feels that it is familiar with a situation, it acts based on its current understanding of the dialog. When faced with an unfamiliar scenario, the agent asks its human user what he or she would do in the agent's situation—advice that we believe is relatively easy for humans to give. Our approach both avoids a training period of constant questioning and allows the agent to discover the consequences of a poor decision without actually making mistakes. We demonstrate our approach both in simulation and on a dialog manager for a robotic wheelchair application.

Introduction

Spoken language allows for natural human-robot interaction, and the ability for a robot to take verbal commands can be especially useful when interacting with those who have limited mobility. The role of a dialog management system is to take dialog from a user—in our case, output from a voice recognition system—and interpret it to determine what action (if any) to take in response. In our work, we focus on a dialog manager for a robotic wheelchair (see Figure 1). The dialog manager's goal is to discover where the user wishes to go and command the wheelchair's navigation software to take the wheelchair to the desired location.

While navigating to a given location may seem to be a well-defined task, several factors make the dialog management challenging. First, the voice recognition system is often noisy—for example, the system may hear the words "coffee machine" when the user asks to go to "copy machine". Even with perfect voice recognition, ambiguities may occur when people use different names for the same location (such as "my desk" and "my office"). Users may also use the same word to refer to multiple locations (such as "elevator" when there are multiple elevators). Finally, to make decisions under uncertainty, the dialog manager must understand the user's preferences: How tolerant is the user of mistakes? How likely is the user to be frustrated by additional questions?

Figure 1: Our dialog manager allows for more natural human communication with a robotic wheelchair.

A good dialog manager must trade between asking questions to reduce its uncertainty (thus avoiding errors), and fulfilling the user's request within a reasonable amount of time. Partially Observable Markov Decision Processes (POMDPs) provide a theoretical framework for making decisions under uncertainty and have been successfully applied to dialog management situations. The ability to manage dialog uncertainty has made POMDPs attractive in assistive health-care (Roy et al., 2000; Hoey et al., 2005) and dialog management domains (Williams and Young, 2005; Litman et al., 2000), where the agent must reason about how to respond to user requests. Unfortunately, such real-world problems typically require a large number of parameters that are difficult to specify *a priori*.

One way to handle the problem of specifying the parameters corresponding to vocabulary, word error rate, user preference, etc. is to learn the model parameters online. In particular, we have shown previously that reinforcement

learning can be an effective way to learn dialog models online while interacting with users (Doshi and Roy, 2007a). Reinforcement learning is a form of learning in which the agent receives numeric feedback (or "reward") after every action. The agent adjusts its actions based on the feedback, and, over time, it learns how to maximize the reward it expects to receive.

While the reinforcement learning approach has been demonstrated in a wide variety of problems, including human-robot interaction (Litman et al., 2000), it has not met with widespread adoption in dialog management systems for several reasons. First, requiring the user to supply reward feedback after each action may be tedious, leading to frustration and inaccurate results. Second, in the reinforcement learning framework, the dialog manager will only learn about the consequences of a poor decision after making a mistake and experiencing a large negative reward. Experiencing a large penalty allows for rapid learning but can quickly lead to user dissatisfaction with the overall system. Finally, humans are notoriously bad at giving accurate numerical feedback which can cause the system to learn to do the wrong thing.

In this work, we present an alternative approach to online learning in human-robot interaction in which we learn a POMDP model online from data and use that model to derive a correct interaction strategy. By building an explicit model, the interaction agent can both assess its confidence in its own decision making and decide when additional training is needed. Instead of a reward signal after each interaction, we propose the concept of a meta-query, that is, a question about an action that the agent should take. These meta-queries take an intuitive form:

> "I think you definitely want me to go to the printer. Should I go to the printer?"

The agent uses these queries to learn about the user's preferences (for example, risk aversion) as well as discover information about their word choice and voice recognition noise. The agent asks a meta-query only if it is sufficiently confused about what action to take next. This active learning scheme limits the amount of feedback that is required, easing the training burden on the user. We show that such a system can adapt to users in a real robotic wheelchair application.

The remaining sections are organized as follows: Section I describes the basic POMDP dialog model and Section II describes how we incorporate the unknown model parameters into a larger POMDP. We present our algorithm in Section III and results in Section IV. Sections V and VI summarize our results and relate them to other work in POMDP model learning.

I. The POMDP Model

Formally, a POMDP consists of the n-tuple $\{S, A, O, T, \Omega, R, \gamma\}$. S, A, and O are sets of states, actions, and observations. In our wheelchair command-and-control scenario, the states represent locations to which the user may wish to go. The user's desired location cannot be directly observed and must be inferred from a set of noisy observations—in our case, keywords from a voice recognition system. The actions represent physical locations to which the wheelchair may drive, as well as questions that the wheelchair may ask the user. Figure 2 shows a cartoon of a simple dialog model.

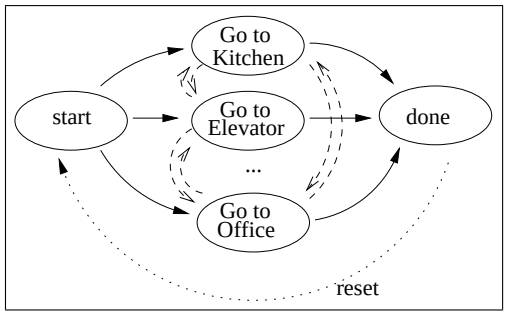


Figure 2: A toy example of a dialog POMDP. The nodes in the graph are different states of the dialog (i.e., user intents). Solid lines indicate likely transitions; we assume that the user is unlikely to change their intent before their original request is fulfilled. The system automatically resets once we reach the end state.

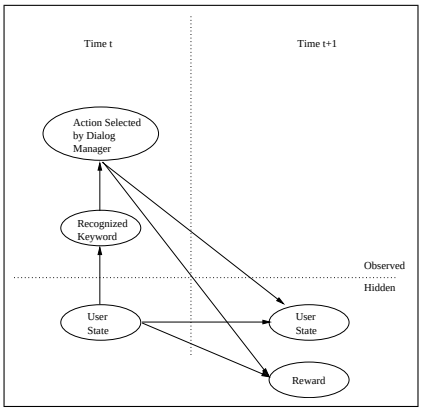
The transition function $T(s'|s,a)$ is a distribution over the states to which the agent may transition after taking action a from state s. Similarly, the observation function $\Omega(o|s,a)$ is a distribution over observations o that may be seen in state s after taking action a. If the observations are keywords, for example, the observation model might encode that the keyword "coffee" is commonly heard when the user wishes to go to the coffee machine. The reward function $R(s,a)$ specifies the agent's immediate reward for each state-action pair. In the wheelchair scenario, the agent may incur a small negative reward for asking a clarification question about where the user wishes to go. Similarly, it may incur a large penalty for taking the user to an incorrect location. Finally, the discount factor $\gamma \in [0,1)$ measures the relative importance of current and future rewards.

Since the true state—the user's intent—is hidden from the agent, it must choose actions based only on past actions and observations. In general, the optimal action to take now will depend on *all* prior actions and observations; however, keeping a history of the entire dialog to date can become quite cumbersome. Fortunately, it is sufficient to store only a distribution over possible user intents—known as a belief—as a sufficient statistic for the past history of actions and observations. If the agent takes action a and hears observation o from an initial belief b, we can easily update the belief using Bayes rule:

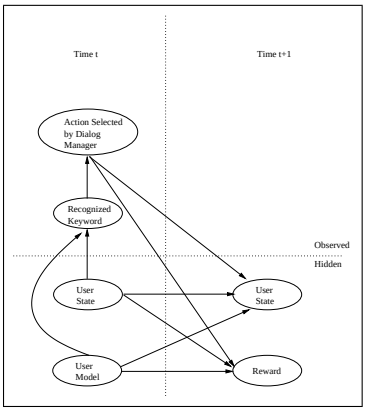
$$b^{a,o}(s) = \frac{\Omega(o|s',a)\sum_{s\in S} T(s'|s,a)b(s)}{\sum_{\sigma\in S}\Omega(o|\sigma,a)\sum_{s\in S} T(\sigma|s,a)b(s)} \quad (1)$$

If the agent has a set of POMDP model parameters that accurately describe the user, then it can simply solve the POMDP for the dialog management policy. The solution

Proc. LangRo'2007
978-972-96895-2-9



(a) Standard POMDP model



(b) Model-Uncertainty POMDP

Figure 3: (a) The standard POMDP model. (b) The extended POMDP model. In both cases, the arrows show which parts of the model are affected by each other from time t to $t+1$. Not drawn are the dependencies from time $t+1$ onwards, such as the user state and user model's effect on the recognized keyword at time $t+1$.

to a dialog POMDP model is a policy that maps beliefs to actions. If the goal is to maximize the expected discounted reward, then the optimal policy can be found by solving the Bellman equations:

$$V^*(b) = \max_{a \in A} Q^*(b,a), \quad (2)$$

$$Q^*(b,a) = R(b,a) + \gamma \sum_{o \in O} \Omega(o|b,a) V^*(b^{a,o}), \quad (3)$$

where the optimal value function $V^*(b)$ is the expected discounted reward that an agent will receive if its current belief is b and $Q^*(b,a)$ is the value of taking action a in belief b. The optimal policy $\pi^* : P(S) \rightarrow A$ can be extracted from the value function using

$$\pi^* = \max_{a \in A} Q^*(b,a). \quad (4)$$

The exact solution to equation 3 is PSPACE-hard but point-based approximations (Pineau et al., 2003) can be used to find high quality solutions efficiently.

II. Modeling POMDP Uncertainty

The problem with using a POMDP to compute a dialog policy is that some of the individual model parameters $\{S, A, O, T, \Omega, R, \gamma\}$ are difficult to specify. It is reasonable to assume that the parameter sets S, A, and O are fixed and known beforehand. For example, in our dialog management task, S could represent all the places that a user may wish to go based on some map initially provided to the robot. The actions A can be pre-specified clarification questions or movements the wheelchair may take, and the observations O the keywords received from a voice recognition system. However, determining the parameters in T, Ω, and R is more difficult, as these parameters describe the user's preferences and the noise in voice recognition system.

However, just as the user's true intent is hidden from the agent, we can also represent the true parameters of the dialog model as hidden variables. We can therefore extend our basic dialog model by including the model parameters as part of the hidden state. We call this new representation a "model-uncertainty" POMDP in which the state space consists of both the user's intent and the true dialog parameters. In this new POMDP model, the state space becomes the set $\tilde{S} = S \times M$, where S is the user space as before, and M is the space of dialog models as described by all valid values for the model parameters. We note that the new state space $\tilde{S}$ is continuous and high dimensional.

Each state $\tilde{s}$ therefore describes a particular user intent s and a particular user model m. The model component m of the state contains the probability distribution describing how the user state s changes, as in the standard POMDP. The observations and rewards received for taking a particular action for a particular user intent now also depend on the hidden dialog model state. To generate policies tractably, we assume that the model component m itself is fixed, that is, the parameters of the user model do not change over time.

Figure 3(a) shows the standard POMDP process. The arrows in the graph show which parts of the model are affected by each other from time t to $t+1$, for instance, the reward at time t is a function of the state at the previous time and the action chosen by the dialog manager. The parameters defining this function are known *a priori* although every part of the model below the "hidden" line is not directly observed by the dialog manager and must be estimated on-line. In contrast, figure 3(b) shows the extended model. The reward at time t is still a function of the state at the previous time and the action chosen by the dialog manager, but the parameters are not known *a priori* and are therefore hidden model variables that must be estimated along with the user state.

Transition and Observation Uncertainty In the previous section, we introduced the belief as a distribution over possible user states. In the model-uncertainty representa-

tion, our belief is now a joint distribution over both the possible user states and the possible user model parameters. Just as we must specify an initial belief over user intents (for example, in Figure 2 we assume that we begin in a "start" state before the user has any intent), we must now specify an initial distribution over possible dialog models—a Bayesian prior on the models. The Bayesian approach is attractive in the dialog setting because we may have strong notions regarding certain parameters, but the exact values for the full set of parameters is typically difficult to specify. For example, we may not know the exact probability of hearing the word "coffee" if the user wants to go to the coffee machine, but we can guess it is probably high. Similarly, we can guess that there is a significant positive reward for driving to the right location and a significant negative reward for driving to the wrong location. We establish a prior distribution over the model parameters to express our domain knowledge, and improve the prior distribution with experience.

The need to represent the prior belief over models raises the question of how to represent this belief. The user state space is a discrete state space, so a standard histogram or multinomial distribution can be used. However, the model parameters such as the transition functions T are continuous parameters of distributions themselves; a distribution over T is effectively a distribution over distributions.

Since T and Ω are collections of multinomial distributions, the Dirichlet distribution is a natural choice of prior. The Dirichlet distribution places a probability measure over the "simplex" of valid multinomials. Figure 4 shows an example of such a simplex for a discrete random variable X where X can have three different outcomes with different probabilities, e.g., $p(X) = [0.25, 0.25, 0.5]$. Each value of $p(X)$ is a different point on the triangular simplex shown in figure 4 and the Dirichlet gives a measure of the likelihood of each such distribution. If $p(X)$ is in fact a transition probability distribution $p(X) = p(\cdot|s,a)$, then each possible transition probability distribution (i.e., each possible user model) is also some point on this simplex, with probability also described by the Dirichlet. As the agent's confidence in a particular model of user behavior increases, the probability mass of the Dirichlet distribution becomes increasingly concentrated around a single point.

Given a set of parameters $\alpha_1 ... \alpha_m$, the likelihood of the discrete probability distribution $p_1 ... p_m$ is given by

$$P(\underline{p}; \underline{\alpha}) = \eta(\underline{\alpha}) \prod_i^m p_i^{\alpha_i - 1} \delta(1 - \sum_i^m p_i),$$

where η is a normalizing constant. The process for updating Dirichlet estimate of the multinomial given additional data is straight-forward. For example, suppose we are given a set of observation parameters $\alpha_1 ... \alpha_{|O|}$ corresponding to a particular *s*,*a*. If we observe observation o_i, then a Bayesian update produces new parameters $(\alpha_1, \ldots, \alpha_{i+1}, \ldots, \alpha_{|O|})$. Thus, we can think of quantity $\alpha_i - 1$ as a count of how many times observation o_i has been seen for the (*s*,*a*) pair. Initially, the expert can specify an educated guess of the distribution—which we take to be the mode of the distribution—and a pre-observation total that represents the expert's confidence in his guess.

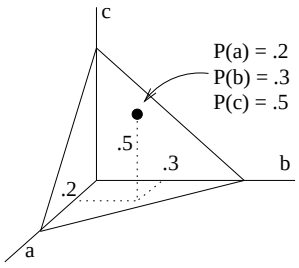


Figure 4: An example simplex for a multinomial that can take three different values (a,b,c). Each point on the simplex corresponds to a valid multinomial distribution; the Dirichlet distribution places a probability measure over this simplex.

Reward Uncertainty Next, we must specify a distribution over rewards. We fix a large positive reward value for driving the user to the correct location, and a small penalty for confirming the correct location with the user (for the minor inconvenience of having to communicate with the robot). These two reward values set a scale for the remaining reward values. We assume that the reward values are uniformly distributed between these ranges. The ranges are expert-specified initially, but the range shrinks as the model of user preferences becomes increasingly certain.

Passive Model Learning The Dirichlet transition, observation and uniform reward priors together specify a distribution over possible POMDP models. The agent can learn some information about the model through user interactions and improve the certainty of the model distribution. For example, suppose that the agent initially hears the word "printer," and user responds to the affirmative when the agent asks if the user wishes to go to printer. Then the agent can increase the probability that word "printer" is associated with the printer location. However, if the user responds to the negative, then the agent can infer that either the word "printer" is not associated with the location printer, or that printer is a commonly the output of a voice recognition error. Likewise, the agent can discover what are the most popular places where the user wishes to go (information about the transition model).

Active Model Learning Other information cannot be learned through user interactions. If the agent is only listening for location keywords, it cannot determine the user's frustration due to a poor action or repeated questions. One option would for the user to input such feedback into the agent; however, even from a small set of user tests in our lab, we found that it was often difficult to explain to users how to input reward values that would lead to the desired behavior from the wheelchair. Such training was also tedious. Thus, we introduced an additional ac-

Proc. LangRo'2007
978-972-96895-2-9

tion to the dialog manager's options: the meta-query. For example, if the wheelchair is fairly certain that the user wishes to go to the printer, it might ask:

> "I think you definitely want me to go to the printer. Should I go to the printer?"

On the contrary, if the wheelchair thinks that the user may want to go to the printer but is not very certain, it might ask:

> "I think you may want me to go to the printer. Should I go to the printer?"

The choice of adverb gives the user an intuitive sense of the agent's uncertainty. Thus, the user can advise the robotic wheelchair based on their internal preferences. For example, if the user is risk averse, they may respond "yes" to the first question but "no" to the second question. If the user answers a question to the negative, the wheelchair might follow up with further questions such as,

> "In that case, I think I should confirm that you want to go to printer first. Is that correct?"

until it receives an affirmative response (assuming that the observation space has been augmented with yes/no keywords)[1]. These meta-queries are not perfect, since the user cannot know the true source of the wheelchair's confusion, but we believe they can provide a more natural way for the human to instruct the robot. We therefore add a set of meta-queries to the action set of the extended POMDP. Each meta-query has a fixed probability of a "yes" or "no" response for each model, which has the effect of changing the model component of the current belief. For simplicity, we fix the cost of each meta-query across all models.

III. Solving the Model-Uncertainty POMDP

Augmenting the original state space with the model parameters provides a principled way of thinking about the actions that result from uncertain dialog models. In Section IIIA, we validate our approach in simulation by solving this model-uncertainty dialog model directly when only a few discrete parameters are unknown. Unfortunately, the increase in the size of the state space also leads to computational intractability; in Section IIIB, we present an approximation that allows us to scale to real-world problems.

IIIA. Solving the Model-Uncertainty POMDP directly

In general, the parameters transition, observation, and reward functions are continuous-valued, with an infinite number of possible models. As such, the model-uncertainty POMDP is especially difficult to solve using standard methods. In special situations, however, uncertainty in the dialog model may be expressed as a small, discrete set of possible models rather than a continuous

[1] In our tests, we used an abbreviated form of the meta-queries for simulation speed.

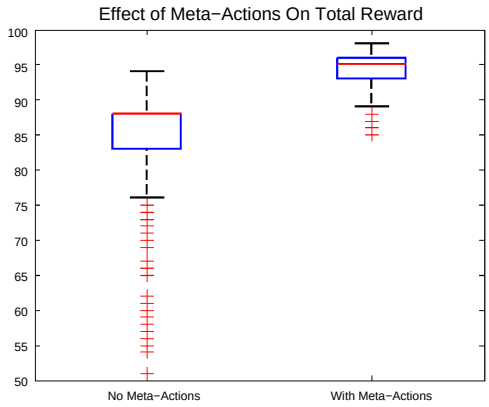


Figure 5: Boxplot of dialog manager performance with a discrete set of four possible models. In this case, the user is very intolerant to errors, but the learner does not initially know this. Although the medians of the two policies are not so different, the active learner (right) makes fewer mistakes than the passive learner (left), leading to overall much less user-annoying behavior.

distribution, making the model-uncertainty POMDP much easier to solve.

For example, consider a scenario where we already have accurate transition and observation models (say, from some prior work with the voice recognition system), but a new user's preference model is unknown. The user's exact preference model may not matter as long as the dialog manager has roughly the appropriate pattern of behavior. In an extreme case, we may decide to only characterize the user's frustration with an incorrect movement as *low* or *high*, and similarly characterize the user's frustration with an incorrect confirmation as *low* or *high*. The user model can be described by two variables $WrongMovePenalty$ $WrongQuestionPenalty$. The two variables $WrongMovePenalty$ and $WrongQuestionPenalty$ can each take either values of *high* or *low*, so that the model for a particular user might be $m =< WrongMovePenalty = high, WrongQuestionPenalty = low >$. This particular user would be conservative, with a preference to be asked questions repeatedly rather than risk being taken to the wrong location. With only four possible dialog models, the state space is still discrete and small, and we can now solve the model-uncertainty POMDP using a standard algorithm (Pineau et al., 2003).

We show simulated results with this very simple scenario of only four possible preference models in Figure 5[2]. The figure compares the performance of the policy without using meta-queries (left column) to the performance of the policy *with* meta-queries. As expected, the system which has the ability to ask meta-queries can use the questions to gain information about the user's internal preference model. It is able to discern that the user is very sensitive about incorrect movements, and therefore it asks more confirmation questions before taking an action. While the

[2] This work previously appeared in (Doshi and Roy, 2007b).

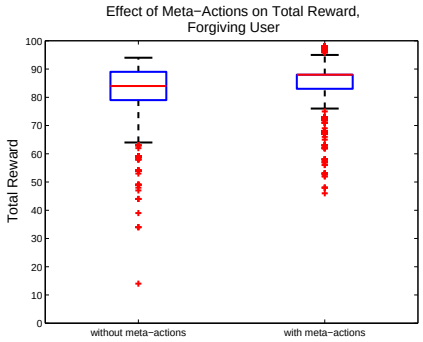


Figure 6: Boxplot of dialog manager performance with a discrete set of four possible models, lenient user. The effect is not as dramatic, but again, the learning dialog manager is able to adapt to the user's preferences and outperform the non-learner, especially in avoiding large mistakes.

difference in medians is not extreme, the reduction in large negative mistakes is substantial—which is particularly important in dialog management, where users will likely find a system that regularly makes mistakes annoying.

In Figure 6, we see similar improvements for the scenario where the user is fairly tolerant to mistakes. Again, the learning dialog manager outperforms the non-learner because it is able to determine the user's internal preference model and therefore ask fewer confirmation questions before acting.

Unfortunately, our approach of representing the user model as discrete values (such as $WrongQuestionPenalty = low$) does not scale well. Experimentally we found that even a modest increase in the number of possible user models from 4 to 48 meant that the model-uncertainty POMDP could no longer be solved using standard solution techniques. While it may be possible to group the possible combinations of user preferences into a few representative models (since the effects of small changes to the preference model may not be apparent to the user), discretizing other parts of the user model such as vocabulary choices quickly produces an exponential number of states. For example, for each keyword the user might utter, we have to consider how likely it is to be heard in each goal location. We therefore turn to approximation techniques which will allow us to represent a larger class of models with continuous parameters.

IIIB. Approximately Solving for a Dialog Policy

Instead of trying to solve for a dialog policy that incorporates both the uncertainty of the user model and the uncertainty of the user state, we separate the problem into two parts: first, we use the current belief over models to establish a representative set of candidate dialog models, and we solve for the optimal policy in each model. We then use these models to choose an action that has minimal risk; if the risk of all other actions is greater than the cost of asking a question for all possible models, we ask a meta-query to improve our estimate of the true user model and reduce the risk of errors. As we interact with the user, we update our collection of possible dialog models to reflect our changing belief about the user model. Table 1 outlines our approach for the continuous dialog parameter case.

Table 1: Dialog model learning approach using Bayes risk and meta-queries.

DIALOG MODEL LEARNING WITH BAYES RISK

- Sample POMDPs from a prior distribution over dialog models.
- Interact with the user:
  - Use the dialog model samples to compute the action with (approximately) minimum Bayes risk.
  - If the risk is larger than a given ε, perform a meta-query.
  - Update each dialog model sample's belief based on the observation received from the user.
- Periodically resample from an updated prior over dialog models.

Minimum Risk Action Selection If we know the correct user model m, then the optimal action to take (either confirming the user intent, or executing an action) is a_{opt}. Let us define a loss function $L(a, a_{opt})$, which describes the cost of taking a different action a. If we know the correct model m, we can solve the model and compute the value of each action a using a standard POMDP solution algorithm to solve equation 2. The loss function of can then be calculated as $Q(b,a) - Q(b,a_{opt})$, where a_{opt} is the optimal action.

We cannot calculate $L(a, a_{opt})$ since we do not know m, but we do have a belief $p_M(m)$ over models that allows us to calculate the expected loss $E_M[L]$. This expected loss is also known as the "Bayes risk":

$$BR(a) = \int_M (Q_m(b_m, a) - Q_m(b_m, a_{opt,m}))p_M(m), \quad (5)$$

where M is the space of dialog models, b_m is the current belief over possible user intents according to dialog model m, and $a_{opt,m}$ is the optimal action for the current belief b_m according to dialog model m. Let $a^* = \arg\max_{a \in A} BR(a)$ be the action with the least risk. If the risk $BR(a^*)$ is less than fixed cost of a meta-query, that is, if the least expected loss is still more than a certain threshold, we perform the meta-query, otherwise we choose the action a^*.

Intuitively, equation 5 computes the potential loss due to taking action a instead of the optimal action a_{opt} according to dialog model m and weights that loss by the probability of model m. When we are sufficiently sure about the model, the risk will be low; when we are unsure about the model, the risk may be high but the series of meta-queries will lead us to choose the correct action and avoid the risk. We unfortunately cannot solve equation 5

Proc. LangRo'2007
978-972-96895-2-9

exactly because the integral is over the model parameters, and the solution would require us to solve for the value functions of an infinite number of POMDPs. However, we can use numerical techniques to find an approximation. Our belief over user states and user models gives us the probability of each model $p(m)$; if we draw sample models from this distribution, we will draw many samples in regions where $p(m)$ is high and few samples from where $p(m)$ is low. The more samples we draw, the better the densities of the samples will represent the distribution from which they were drawn. Thus, we can approximate equation 5 with the sum:

$$BR(a) = \sum_i (Q_i(b_i, a) - Q_i(b_i, a_{opt,i}))w_i, \qquad (6)$$

where Q_i provides the value of taking actions from belief states according to dialog sample i.

By drawing samples from the distribution $p(m)$, we are using the samples to approximate this distribution. However, the distribution over models will change as the wheelchair interacts with the user. The wheelchair must therefore periodically update the sample of dialog models that it is using to approximate its belief over models. If model samples are drawn from the current distribution over models, the weight w_i of each model is simply $\frac{1}{N}$, where N is the number of samples. However, for computational reasons—since we must solve every dialog model that we sample—it may be undesirable to resample models every time some new information changes our belief over possible models. In this case, the original sample set of models can be re-used by changing the weight of each model and representing the distribution $p(m)$ as a set of weighted samples. At each time step, the weight of each model should be adjusted to be proportional to the ratio of the previous likelihood of the sample and its likelihood given new information. While it is possible to provide formal bounds on the number of samples needed to approximate the Bayes risk to a specified degree of accuracy, these bounds are loose and in practice we found that fifteen samples sufficed for our dialog management application.

POMDP Resampling In some cases, the set of weighted samples may no longer accurately represent the true distribution over models, requiring a new set of sample models to be generated. The need for resampling may arise because one of the models becomes far more likely than the other dialog models in our sample set. If one model's weight w_i is close to 1, and the rest are close to 0, then the risk will appear to be quite small. This approximation is reasonable when the risk is truly small, but we do not want the dialog manager to become over-confident due to a poor set of candidate models. Another reason to resample models is that an interaction may have provided information that made all of the models in our current set very unlikely, and we would like our sample set to reflect our current belief over the dialog parameters.

We have two sources of information when it is time to update our sample set of dialog models. One source is the history of the most recent dialog, which consists of action-observation pairs $h = \{a, o\}$. Another source is the set of meta-queries $Q = \{(q, r, h')\}$, where h' is the history of the dialog from the initial belief to the query, q is the query, and r is the user's response to the query. Given h and Q, the posterior probability $p_{M|h,Q}$ over models is:

$$p_{M|h,Q}(m|h, Q) = \eta p(Q|m)p(h|m)p_M(m), \qquad (7)$$

where η is a normalizing constant. Note that if p_M is a Dirichlet distribution, then $\eta' p(h|m)p_M(m)$ is also a Dirichlet distribution since the likelihood $p(h|m)$ is product of multinomials. Recall that updating the Dirichlet distribution corresponded to adding counts—for example, if wheelchair observed the word "printer" after asking the user where he wished to go when the user truly wished to go to the printer, then we would add 1 to the Dirichlet parameter for hearing "printer" given a general query, true user goal is printer. The trouble is that we never know the true user state—we only have actions and observations.

Given a complete dialog, however, and assuming that it is unlikely that the user switched their objective in mid-dialog, it is possible to accurately infer the most likely underlying states from a history of actions and observations using the standard forward-backward algorithm. We can use the output of this algorithm to update the Dirichlet counts. This is a modified form of the standard Expectation-Maximization algorithm, and thus the prior will converge to some local optimal dialog model.

Incorporating meta-query information requires a different approach, since each specific meta-query response provides information about the dialog policy, not the dialog model parameters. We do not have a closed-form expression for $p_{M|h,Q}$, so we must use sampling to draw dialog model samples that are consistent with all of the meta-queries that have been asked so far. Each query in the set Q provides a constraint on the feasible set of dialog models M. Dialog models are feasible if their policy is consistent with the responses in the meta-query.

Computing this feasible set directly is intractable, however, given the set Q, we can check if a sampled dialog POMDP is consistent with the previous meta-query responses stored in Q. Thus, to sample POMDPs, we first sample dialog POMDPs from the updated Dirichlet priors. Next, we solve for the optimal policy of each model (which can be done quickly, since each dialog model sample is discrete and relatively small) and check if each dialog model's policy is consistent with the previous meta-query responses stored in Q.

IV. Results

Simulation Figure 7 shows results from a simulated dialog manager for our wheelchair application. The states consisted of locations where the user wanted to go, and the observations consisted of keywords extracted from utterances. Actions included open-ended questions, confirming a particular state, and driving to a particular location. Above, we see the usefulness of the Bayes-risk approach (compared to stochastic actions selection based on the weights of the sampled models) when the reward

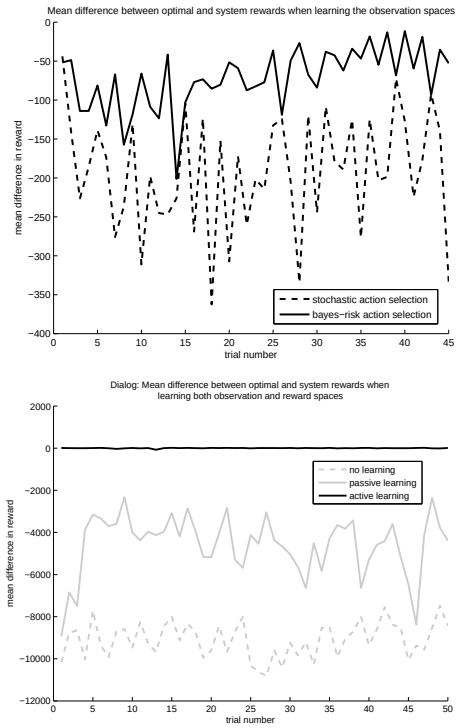


Figure 7: Dialog manager simulation results. Top: results from learning only the observation model. Bottom: benefits of active learning when learning both the observation and reward model.

model is known. In this case, the Bayes risk action selection allows us to choose non-risky actions.

The usefulness of our approach is even more dramatic when the reward prior is uninformative (below). In this case, the dialog manager can improve somewhat by passively updating its priors based on what it has heard (solid gray line). However, simply listening cannot provide the dialog manager information about the user's preferences. Moreover, since the active learning system asks the user for help whenever it is confused, this system does not suffer from an initial dip in performance before the model estimate converges to the true model. The meta-queries allow the active learner to learn while avoiding mistakes, thus maintaining a high level of performance.

Robotic Wheelchair We also validated our approach on a dialog manager for a robotic wheelchair with a simple user study. The underlying POMDP, with 10 states, 38 observations, and 21 actions, used keywords from a voice recognition system output as observations. Initially, each state had one strongly mapped observation (such as 'printer' for the printer location). The remaining observations received uniform initial priors. Four users conducted 12-15 interactions (20-25 minutes) with the system.

By asking meta-queries, the dialog manager was able to successfully complete all 57 interactions without making a serious error, that is, trying to drive the user to an incorrect location. Table 2 shows that the proportion of dialogs with meta-queries decreased significantly from the

Table 2: Proportion of dialogs with meta-queries by location. The decrease in the number of meta-queries from the second to the third time the location was asked for is statistically significant at the $p = 0.05$ level.

First half of interactions	.79
Second half of interactions	.48

Table 3: Proportion of dialogs with meta-queries. The decrease in the proportion of queries is significant at the $p = 0.05$ level.

How often place was requested	First time	Second time	Third Time
Total number of requests	29	15	8
Number of requests with meta-queries	22	11	2
Proportion of requests with meta-queries	.76	.73	.25

user's initial interactions with the system to his or her final interactions with the system. The proportions are relatively high because the users, being new with the system, asked for a variety of locations or experimented with different vocabulary. In Table 3, we see that the system often asked a meta-query the first or second time a user asked to go to a particular location, but there was a significant drop in the number of meta-queries after the third time the user requested to go to the same place. We note that our active-learning approach allowed our dialog manager to automatically determine the length of its training phase. Our trials were too small to draw strong conclusions from these results (especially since we were unable to account for the effect of the novelty factor in the interactions), but our quantitative analysis does show statistically significant decreases in the number of meta-queries asked as the dialog manager adapts to each user.

Next we present some sample dialogs to illustrate our system's learning. Table 4 contains a sample dialog where the system used a meta-query to establish the meaning of a new word. In the initial dialog, the word 'forecast' is new to the system, that is, it has no prior on what that word may mean. The system determines that the Bayes risk is great enough to warrant asking for a policy clarification. As a result, future dialogs are biased toward choosing POMDP samples that satisfy the initial policy query request, which allows later dialogs to avoid the same kind of training.

In Table 5, the system adapts to a situation where two words, 'coffee' and 'copy' are often mistaken for each other in the voice recognition system. We stress that at the beginning of the dialog, the system was initialized to believe that the word 'coffee' was most likely if the user wished to go to the coffee machine, while the word 'copy' was most likely if the user wished to go to the copy machine. The other key words—"tea" and "kitchen"—-in this segment began with uniform priors.

During the first conversation, the agent is confused by the many new words. Even after asking a meta-query,

Proc. LangRo'2007
978-972-96895-2-9

Early Conversation
ROBOT: How can I help you?
USER: Give me the forecast.
ROBOT: I'm confused. What action should I take now?
The system pops up a meta-query window, and user indicates action "show weather" is correct.
ROBOT: Showers.
Later Conversation (with the same user)
ROBOT: How can I help you?
USER: What's the forecast for today?
ROBOT: Do you want the weather?
The system has partially the word "forecast" and confirms the correct state.
USER: Yup.
ROBOT: Showers.

Table 4: Learning a new word: The system learns the meaning of a new word through a meta-query.

it is confused by hearing the word "copy" since the user seems to want to go to the kitchen. After several interactions, the agent goes to the correct location. We note that even in this tricky scenario, the use of the meta-queries helped the agent focus on to the kitchen when the only word it recognized—"copy"—seemed to indicate that the user wished to go to another location. In the later dialog, we see the effect of the learning: upon hearing the words "kitchen" and "copy," the agent correctly infers that the user wishes to drive to the kitchen.

V. Discussion and Related Work

Prior work in POMDP learning has also considered incorporating the parameters of the POMDP as additional hidden states into a larger model-uncertainty POMDP. One approach, the Beetle algorithm (Poupart et al., 2006) solves the problem analytically by casting an MDP with unknown parameters as a POMDP. However, this approach would be difficult to extend to the POMDP case as the state space would now be doubly continuous and very high dimensional. Note that our goal is policy learning, rather than explicitly learning a vocabulary as in (Lopes and Teixeira, 2000; Lopes and Chauhan, 2007). We learn additional word models only as a function of how they affect the policy.

An approach that is closer to ours is the Medusa algorithm (Jaulmes et al., 2005). The Medusa algorithm also samples POMDPs from a prior over model parameters, but the actions are chosen stochastically and queries are made to a state oracle, not for policy information. While the Medusa approach can allow for faster learning, our action selection criterion is more robust (something valuable for dialog management applications with a robotic wheelchair) and our queries provide a more natural way for the user to provide feedback to the robot. To our knowledge, this is the first example of using Bayesian reinforcement learning on a real-world application.

Early Conversation
ROBOT: How can I help you?
USER: I want some tea from the kitchen.
ROBOT: How can I help you?
USER: The tea is in the kitchen near the coffee machine.
The system heard "copy" instead of "coffee" and is confused about the new word "kitchen."
ROBOT: How can I help you?
USER: Go to the kitchen.
ROBOT: I'm confused. What action should I take now?
The user indicates "Confirm the coffee machine area" is the correct action.
ROBOT: Do you want to go to the coffee machine area?
USER: Yup.
… *The system asks a few more (non-meta) questions before re-confirming the coffee machine state.*
ROBOT: Do you want to go to the coffee machine area?
USER: Yup.
ROBOT: Going to the coffee machine area.
Later Conversation (with same user)
ROBOT: How can I help you?
USER: Take me to the kitchen.
The system hears a noisy combination of "nick" and "kitchen" (mostly "kitchen").
ROBOT: How can I help you?
USER: I want some coffee.
The system hears "copy" instead of "coffee," but with its learned observation model, it confirms correctly.
ROBOT: Do you want to go to the coffee machine area?
USER: Yup.
ROBOT: Going to the coffee machine area.

Table 5: Learning about observation noise: the system learns new words for a noisy pair of states.

VI. Conclusion

We developed an approach for active learning for dialog managers that robustly adapts to different users. Meta-queries—questions about actions that the agent is thinking of taking—and a risk-averse action selection criterion allowed our agent to behave robustly even when its knowledge of the POMDP model was uncertain. Moreover, we demonstrated that this model could be successfully transferred to a dialog manager on a robotic wheelchair.

References

Doshi, F. and Roy, N. (2007a). Efficient model learning for dialog management. In *Proceedings of Human-Robot Interaction (HRI 2007)*, Washington, DC.

Doshi, F. and Roy, N. (2007b). Efficient model learning for dialog management. In *Technical Report SS-07-07*, Palo Alto, CA. AAAI Press.

Hoey, J., Poupart, P., Boutilier, C., and Mihailidis, A. (2005). Pomdp models for assistive technology. *IATSL Technical Report*.

Jaulmes, R., Pineau, J., and Precup, D. (2005). Learning in non-stationary partially observable markov decision processes. ECML Workshop.

Litman, D., Singh, S., Kearns, M., and Walker, M. (2000). NJFun: a reinforcement learning spoken dialogue

system. In *Proceedings of the ANLP/NAACL 2000 Workshop on Conversational Systems*, Seattle.

Lopes, L. S. and Chauhan, A. (2007). How many words can my robot learn? an approach and experiments with one-class learning. *Interaction Studies*, 8(1):53–81.

Lopes, L. S. and Teixeira, A. (2000). Human-robot interaction through spoken language dialogue. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, pages 528–534.

Pineau, J., Gordon, G., and Thrun, S. (2003). Point-based value iteration: An anytime algorithm for pomdps. *IJCAI*.

Poupart, P., Vlassis, N., Hoey, J., and Regan, K. (2006). An analytic solution to discrete bayesian reinforcement learning. In *ICML*, pages 697–704, New York, NY, USA. ACM Press.

Roy, N., Pineau, J., and Thrun, S. (2000). Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting of the ACL*, Hong Kong.

Williams, J. and Young, S. (2005). Scaling up pomdps for dialogue management: The "summary pomdp" method. In *Proceedings of the IEEE ASRU Workshop*.

Robots that inhabit the human world

Carlos Herrera[1], Rob Clowes[2], T.M. McGinnity, [1] Tom Ziemke[3]

[1] Intelligent Systems Research Centre, University of Ulster
[2]COGS, University of Sussex
[3]University of Skövde, School of Humanities and Informatics, Sweden

c.herrera@ulster.ac.uk

Abstract

Recent trends in cognitive science stress situatedness and adaptivity as fundamental characteristics of autonomous embodied agents. Largely steaming from the critique to classical AI and notions of representation, these new trends often ignore the cultural environment, including articulated language, assuming the continuity between animal and human forms of cognition. Nevertheless, if robots are to inhabit a human world, they need to be situated in such a linguistic environment in a way that surpasses animal forms of cognition. This pressing requirement may in turn have consequences in the development of how we understand autonomy and the self.

1. Introduction

In this paper, we investigate theoretically what it means for robot developers to integrate robots in a linguistic human niche. We consider language to be an existing social structure – therefore a human niche is an environment which includes a social structure articulated as a public language. To be situated in such an environment is to have the cognitive capacity to understand other's utterances, communicate one's beliefs about oneself and the world, respond to imperatives and motivate actions in others, etc …

Traditionally, cognitive science has considered language as a tool for communication between fully-fledged cognitive systems. On this view, 'private language' is the privileged internal representational structure in which mental processes are carried out. Public language is the imperfect translation of what goes on inside. Supracommunicative theories (Clark, 1998) on the other hand, consider the existence of public language to be an essential configuration of the niche an agent needs to adapt to, and which may shape functional aspects of such 'cognitive machinery'. In other words, it may play a role in shaping and distributing cognitive problems among individuals in a social environment and within the agent itself.

Even though language processing was one of the major topics in the so called traditional Artificial Intelligence, emphasis on adaptation and sensory-motor coordination has driven the focus of research into the challenges of embodied action. Nevertheless, most robotic applications envisioned for the near future involve linguistic interaction with humans. Understanding how being embedded in a linguistic environment modifies the cognitive problems a robot faces is a path this kind of robotic technologies will have to walk down sooner or later.

The need for robots that could potentially cohabit with us is argued by some to be pressing (Brooks, 2002). This is the case, for instance, in assistive robotics, in which robots are imagined as valid tools to assist handicapped and elderly people in everyday life. Such robots would need to maintain complex and potentially open-ended interactions with the human-beings and would ideally be able to use language to communicate. One basic feature of an assistant is that it must be able to accept descriptions, commands and instructions. Of course, we cannot disregard the possibility that robots may need to inhabit even more complex socio-linguistic niches narrative niches[1], and perhaps in the process escape their condition as machines.

2. Cognitive linguistic agents

There are a few hints on how cognitive problems are shaped by the existence of a linguistic niche. Clark, for instance, offers a number of potential ways in which public linguistic artefacts can transform cognitive problems, such as memory augmentation, environmental simplification, coordination and the reduction of on-line deliberation, taming path-dependent learning, attention and resource allocation or data manipulation and representation (cf Clark, 1998). Arguably, the consideration of public language can also help us better understand important issues such as the symbol-grounding problem.

The consequences being embedded in a linguistic environment have for an adaptive agent are ultimately dependant on how behaviour is controlled and organised. We consider the framework of neuro-robotics. In classic AI, control is conceived as intrinsically semantic. Public language only emerged as the externalization of internal control. Neural controllers, on the other hand, are naturally subsymbolic and the patterns of activity are defined by dynamic rather than semantic rules. Andy Clark view the brain as an organ engaging in environmental interactions through an iterated series of simple pattern-completing computations (associative engine). As language becomes part of the world out there

[1] A distinction between narrative vs. mere linguistic niches will be discussed further in the full paper.

for the agent, the brain's task is not to store a system of representations and to translate them into public language, but to situate the agent in such a linguistic framework.

Ultimately, the problem of linguistic artefacts such as cognitive assistants is one of design requirements: how socialised do robots need to be? It may be unrealistic, unnecessary and even undesirable to consider the possibility of a robot that can develop a complete understanding of the human niche. Nevertheless *some* language mediated interactions seem necessary if we are to have useful cognitive assistants. Thus, we propose an incremental approach based on the concept of language games, conceptualised by Wittgenstein (1953, p. 3)

We should consider the robot to be a minimally embedded in the linguistic niche if it is able to participate in some of the number of available language games. The benefit of this approach is that linguistic capabilities are not studied in isolation, but within the embodied practice made available by a social environment. This may help us in posing certain questions on the relationship between language and the human niche. Dreyfus, for instance, argues that "To learn a natural language a computer has to have a body; it must be embodied if it is to be embedded" (Dreyfus, 1996,p. 181). Dreyfus has also argued that an emotional background is also required "the socializable entity will have to be self-moving, have feelings and emotions, be able to detect and care about approval and disapproval, and lot more" (Dreyfus, 2000, p. 346).

Language internalisation and cognitive construction may be important here. The use of language to perturb / or construct an ongoing action, and replace scaffolding might be important not just so that a robot comes to share our life-world, but such that it is able to develop its own flexible modes of cognition. It may also transform how we face classical robotic problems, such as attention, control and the structure of memory.

An interesting further angle would be to examine what new questions would emerge as a result of robots that might substantially internalise language games or cohabit in our niche. Especially if they could use advanced socio-linguistic modes such as narrative[2]. Dennett (1992) suggests that the self is a *centre of narrative gravity*. If selves are complex narrative posits, as this idea suggests, it is reasonable to suspect interesting moral and practical questions to arise as agents are able to participate in and internalise more complex language games.

If we are to build robots that can more capaciously inhabit the human cognitive niche then it seems appropriate to consider these questions in more detail.

[2] And there are reasons to suspect that a valuable care-giving robot might need at least some minimal recognition of narrative.

References

Brooks, R. (2002). *Robot: The Future of Flesh And Machines*. Cambridge, Massachusetts: Allen Lane: The Penguin Press.

Clark, A. (1998). Magic Words: How Language Augments Human Computation. In P. Carruthers & J. Boucher (Eds.), *Language and Thought. Interdisciplary Themes* (pp. 162 - 183). Oxford: Oxford University Press.

Dennett, D. C. (1992). The Self as a centre of Narrative Gravity.

Dreyfus, H. (1996). Response to my critics. *Artificial Intelligence, 80*(1-2), 171–191.

Dreyfus, H. (2000). Response to Collins. In M. Wrathall & J. Malpas (Eds.), *Heidegger, coping, and cognitive science: Essays in honor of Hubert L. Dreyfus* (Vol. 2, pp. 314–349). Cambridge: MIT.

Wittgenstein, L. (1953). *Philosophical Investigations*. New York: Macmillan.

Early Speech Development of a Humanoid Robot using Babbling and Lip Tracking

Jonas Hörnstein[1], Cláudia Soares[1], José Santos-Victor[1] and Alexandre Bernardino[1]
[1]Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal
jhornstein@isr.ist.utl.pt

Abstract

In this work we show how a humanoid robot can learn to produce and recognise both vowels and consonants using a unified method for speech production and recognition. The method is inspired by the motor theory and the discovery of mirror neurons. Both auditory and visual information is used and mapped to the robot's articulatory space where the recognition and speech production is performed. A combination of babbling and imitation is used to learn the maps. We find that the visual information can be useful not only to increase the recognition rate of already learnt phonemes, but also to drive the learning of new phonemes.

INTRODUCTION

Language acquisition is a complex and highly social process. To interact with humans using speech a robot need to be able both to produce and to recognize a number of phonemes. Speech production, speech recognition, and learning of phonemes are usually handled by different processes, but here we handle these tasks with a unified approach. This approach is based on our earlier work (Hörnstein and Santos-Victor, 2007), where we map the speech signal to motor representations in the robot's vocal tract and perform both speech planning and speech recognition in motor space. A similar approach is taken in (Kanda and Ogata, 2007). The idea to use motor space rather than directly using the speech signal comes from the Motor Theory (Liberman and Mattingly, 1985). They found that being able to produce a certain sound also increased the possibility to recognize the same sound. In an other work it has been found that there is an increased activity in the tongue muscles when listening to words that requires large tongue movements (Fadiga et al., 2002). Both these works lead us to believe that the motor area is involved not only in speech production, but also in speech recognition.

In this work we further extend and develop our unified approach by including visual input in the form of a lip tracker and a self clustering algorithm that automatically groups learned motor positions into phonemes. We also show how a humanoid robot can use the described approach to learn both vowels and simple consonants during its early speech development. The robot used in this work is the iCub, Figure 1. The iCub is equipped with sensors in the form of microphones and cameras, and can produce sound through a simulated vocal tract. It has no preprogrammed knowledge about language. Instead it has to learn how to speak by exploring its vocal tract and learn its initial sensory-motor maps using babbling. It also has to learn which sounds are useful for communication with humans, group these sounds into phonemes, and to recognize the same phonemes when pronounced by different speakers. The set of sounds considered as useful depend on the cultural environment in which the robot is placed and therefore has to be learned through the interaction with humans. Here we use different types of imitation games to allow the robot to learn new phonemes and gain speaker invariance.

The rest of the paper is organized as follows. In section 2 we give an overview of the architecture used and especially focus on the new parts like the lip tracker and the clustering algorithm. In section 3 we describe the babbling and imitation behavior that the robot uses to develop its speech. In section 4 we show some experimental results and conclusions are given in section 5.

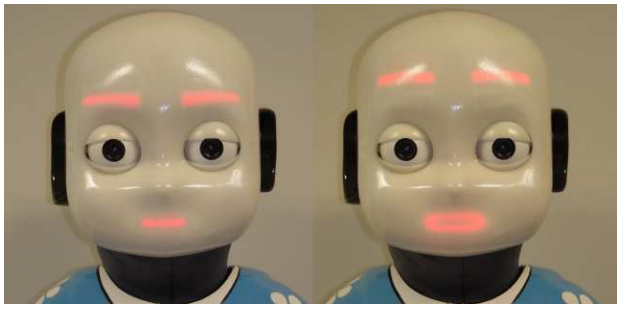


Figure 1: iCub robot learning to speak

System architecture

The architecture used in this work is an extension of the architecture described in (Hörnstein and Santos-Victor, 2007). As in the previous work the architecture consists of a speech production unit, a sensor unit, a sensor-motor map and a speech recognition unit, Figure 2. The main difference compared to the older version is the addition of a visual sensor in the sensor unit and a vision-motor map in the sensor-motor map unit. We have also done some modifications in the position generator that drives the babbling and added a self clustering algorithm in the motor

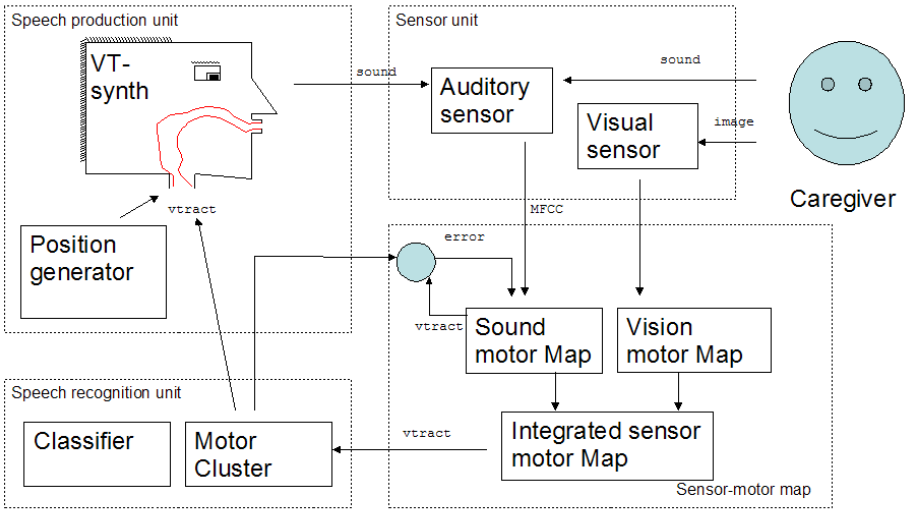


Figure 2: Speech architecture

cluster. In this section we give a short overview of each unit and explain the differences between the current and the previous version in more detail.

Speech production unit

The speech production unit is responsible for moving the lips and producing sound. As in the previous version we do not use a physical model of the human vocal tract, but simulates the vocal tract in a computer model. The model used is vtcalcs developed by Maeda (Maeda, 1990). This model has six parameters that can be used to control the movements of the vocal tract. One parameter is used for the controlling the position of the yaw, one for the protrusion of the lips, one for lip opening, and three parameters for controlling the position of the tongue. A synthesizer converts the vocal tract positions into sound. While the synthesizer works well for vowel-like sounds, it is unable to produce fricatives sounds and can hence only produce a limited set of consonants.

In the new architecture the vocal tract position is also used to control the shape of the robot's lips. Our robot has a very simple lip model consisting of a number of leds that can either show a closed or an open mouth. A simple threshold is used to decide whether the mouth should be shown as open or closed. Examples of the mouth positions are shown in Figure 1.

The most important difference in the speech production unit is the new position generator. While the previous version only created random positions for the vocal tract the new unit offers more advanced babbling behavior. One of the problems with the random position generator was that it created lots of non-humanlike sounds that aren't useful for human-robot interaction and slows down the learning process. In (Soares and Bernadino, 2007), it has been shown that a convex combination of three corner vowels [i], [a] and [u] is able to produce the complete vowel space. The corner vowels represent extreme placements of the tongue and can therefore be considered as known stable points when starting the exploration of the articulatory space.

Thus, in this work we include these corner vowels as starting points in the motor cluster, even though we have previously shown that it is possible to learn those using random babbling. The position generator creates a new sound by picking two positions in the motor cluster and creating a trajectory between those. As shown in (Soares and Bernadino, 2007) we always create tangible speech as long as we stay within the convex envelope of the corner vowels. However, as this would also restrict us to the vowel space we add some noise to the positions before creating the trajectory. This way we allow the robot to also explore the articulatory space beyond the vowel space.

Sensor units

We use two sensors, an auditory sensor unit and a visual sensor unit that extract features from the acoustic and visual spaces respectively. The auditory sensor remains unchanged. A microphone is used to record the sound. The sound is windowed into 30 ms frames and Mel frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980) are calculated for each frame. The visual sensor unit is, on the other hand, a complete new unit that has not been presented in the previous work. In the following we explain this sensor unit in more detail.

The purpose of the visual sensor is to provide visual clues on the position of the vocal tract. While there are methods to find the exact contour of the lips, like the usage of snakes or active contour methods (Kass et al., 1987), these methods are typically too complex to use in speech recognition. With no a priori assumption of the shape of the lips the estimation becomes slow and more error prone. Further more, the complexity of the final description makes further data processing costly. For practical applications where we need to track the movements of the

Proc. LangRo'2007
978-972-96895-2-9

lips in real-time, and are interested in some simple feature like the area of the mouth opening rather than the exact contour, we need a compact representation of the lips. In this work we have chosen to represent the lips by an ellipse, which is fitted to the pixels that belong to the lips. The pixels that belong to the lips are found by using color segmentation. The color segmentation can be done in several different ways. It is usual to extract the color from the first frame using the initial position of the lips. In (Tian et al., 1999) the whole color distribution of the lip region is calculated and modelled as a Gaussian mixture and the EM method is used to estimate both the mixture weights and the underlying Gaussian parameters. Here we use a much simpler method and simply model a lip by its redness, where we define the redness as:

$$Redness = R^2/(R^2 + G^2 + B^2)$$

where R, G, and B are the red, green, and blue value of an RGB-image. If the redness of a pixel is above some threshold we define the pixel as a lip. The threshold can be calculated from the initial frame, but we have chosen a fixed threshold of 0.9. As shown in Figure 3, the threshold seems to work well even for different persons. Of course there are other pixels apart from the lip pixels that are classified as red so we need to know the approximate position of the lips and only use those pixels to fit the ellipse. Here we use a face detection algotithm, based on (Viola and Jones, 2001) and (Lienhart and Maydt, 2002). The face detection algorithm not only gives us an initial estimate for the position of the lips, but also gives us the size of the face which is later used to normalize the area of the mouth opening. However, the face detection algorithm is rather slow so the position and size of the head is therefore only calculated once in the beginning of every experiment and the subject with which the robot interacts is assumed to maintain approximately the same distance to the robot during each experiment.

To fit the ellipse to the lip pixels we use a least square method described in (Fitzgibbon et al., 1999). The result is shown in figure 4. We then use the ellipse to calculate the area of the mouth opening. The ratio between the area of the mouth opening, given by the lip tracker, and the area of the face given by the face tracker, is used as a visual feature and is sent to the vision-motor map.

As said before, the face detection is too slow to be useful for tracking the movements of the lips between two frames in the video stream. We therefore use the method suggested by Lien et. al (Lien et al., 1999). They use Lucas-Kanade tracking algorithm (Lucas and Kanade, 1981) to track the movements of the lips between adjacent frames. One problem with the tracking algorithms is that it is sensitive to the initial feature point selection as most points on the lips have ambiguities around the lip edges. Here we solve this by looking for Harris features (Harris and Stephens, 1988) around the lips and use these as initial points that will be tracked. The result gives us a sufficiently good estimate to maintain an initial estimate of the lip position over the video sequences used in our experiments.

Figure 3: Color segmentation

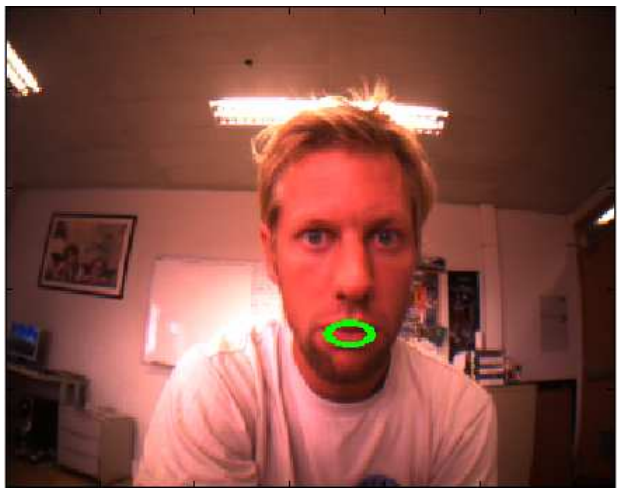


Figure 4: Lip tracking

Sensor-motor maps

The sensor-motor maps are responsible for retrieving the vocal tract position from the given auditory and visual features. We use two separate neural networks to map sound-motor map and the vision-motor map respectively. The sound-motor map is the more complicated of the two, mapping the 12 cepstral coefficients back to the 6 parameters of the vocal tract model. The problem is extra difficult since several positions of the vocal tract results in the same sound, giving several possible solutions for a given set of features. While the position generator described above reduces the risk of producing the same sound from two different positions, we still get some ambiguities that have to be solved through the interaction with a caregiver. For the sound-motor map we use an artificial neural network with 20 hidden neurons.

The vision-motor map is a very simple unit, doing a linear mapping from the mouth opening to the lip height parameter of the synthesizer.

Since the output from both the sound-motor map and the vision-motor map consist of vocal tract positions, the integration of those sensor outputs becomes very simple. Here we simply use a weighted average of the lip height calculated from the two maps. The weight is currently set by hand, but should preferable be set automatically ac-

cording to the quality and intensity of the visual and auditory stimula.

Speech recognition unit

The speech recognition unit contains a motor cluster and a classifier. In our previous version of the architecture the classifier was a simple dictionary that stored motor positions that were considered useful for the communication with the caregiver. In this new version we have implemented an hierarchical clustering algorithm based on (Hastie, 2001), which starts with creating one cluster for each stored position and then iteratively joins the two clusters with the minimum euclidean distance until we only have one single cluster containing all stored positions.

For each level of the clustering process, we have different relationships between data groupings. So, the question is: what is the "natural" grouping for this dataset? To estimate the number of clusters in a data set we used Gap statistic (Tibshirani et al., 2001). This function compares the within-cluster dispersion of our data with that obtained by clustering a reference uniform distribution. This is to compare the gain of raising the cluster number in a structured data with that arising from adding another cluster to a non-informative and not structured set of points.

We choose the first maximum in the Gap statistic as the optimal number of clusters. Each position within the same cluster is considered to be part of the same phoneme or pseudo-phoneme.

The recognition task is handled by the classifier that compares positions given from the sound motor map with the mean positions of each pseudo-phoneme in the motor cluster and can be configured to use either Euclidean distance or the Mahalanobis distance to find the nearest neighbor.

Babbling and imitation

In this this section we describe the mechanisms used by the robot to learn to vocalize vowels and simple consonants. The methods are inspired by the way children develop their speech through a combination of self-exploration in the form of babbling and through interaction with a caregiver. We separate between two types of interactions, the robot imitating a caregiver, and the caregiver imitating the robot. Both these behaviors can be found in the interaction between a child and its parents.

Here we first describe what happens during babbling, then we explain which maps that are updated as the caregiver imitates the robot, and finally what happens when the robot imitates the caregiver. However, we would like to point out that these activities should be seen as parallel rather than sequencial and that all behaviors are active during the whole development.

Babbling

The babbling behavior is realised by the position generator. As explained in the previous section the position generator randomly takes two positions from the cluster, add some noise to the positions, and then create linear trajectory between the two points. In the beginning the noise level is set relatively high in order to explore as much as possible of the articulatory space. With time, and as more positions are stored in the cluster, the noise level in the babbling is gradually reduced and the babbling is focused on the trajectories between the learnt positions.

Each position in the generated trajectory consists of the 6 parameters in Maeda's model. These are then passed on to the speech production unit that calculates the resulting sound. The sound is then fed into the auditory sensor unit that calculates the MFCC and passes these to the sound-motor-map. The sound-motor-map finally tries to map the MFCC back to the original articulator position vector and compares the result with the output from the position generator. The error between the mapped and the correct positions is the used to update the map using a back-propagation algorithm.

There is no update of the vision-motor map during babbling since the robot does not get any visual feedback of its lip position.

Caregiver imitating robot

Having the caregiver imitating the robot is arguable the most important factor in learning both the sound-motor map and the vision-motor map. While the robot can easily learn the map between its own sound and motor positions through babbling, there is a large difference between the speech produced by the robot and normal human speech. The same can be said about a child whose vocal tract is significantly different to that of an adult. Add to that the fact that the sound produced by ourselves is transmitted not only through the air, but also through bone structures in the head which make our own voice sound significantly different compared to the sound produced by others even if we would have exactly the same vocal tract. To compensate for those things we have to interact with other people in our environment and tune the maps according to their voices.

This interaction starts with the robot creating a trajectory in the same way as for the babbling and sending the sound to the speaker. The caregiver then tries to repeat the same utterance with its own voice. It is important that the caregiver repeats the perceived utterance rather than the exact sound produced by the robot. Here we do not handle the problem of deciding whether the person with whom the robot interacts is actually imitating what the robot said or not, but simply assumes that the received response is the same utterance. We also make sure that the utterance has the same length and that it is correctly aligned in time with the utterance of the robot. This is done manually at the moment by selecting some keypoints along the trajectory and finding the same key points in the response of the caregiver. We also extract images from the video stream that match each of the key points.

The maps are then trained using the vocal tract positions of the robot together with the auditory and visual response from the caregiver. The sound from the caregiver is fed

Proc. LangRo'2007
978-972-96895-2-9

into the auditory sensor and the corresponding MFCC are calculated and sent to the sound-motor map. The mapped position is compared to the vocal tract position used by the robot and the map is updated to compensate for the error. In the same way the image of the caregiver is fed into the visual sensor which calculates the mouth opening and sends the result to the vision-motor map. Again the mapped position, this time of the lip height only, is compared to the original position and the map is updated according to the error.

This is repeated for various utterances and preferably with several different caregivers in order to increase the robots posibility to correctly map utterances from other persons to its own vocal tract in order to reproduce the same sound or to recognize what the other person actually said.

Robot imitating caregiver

One problem having the caregiver imitate the robot is that the robot is not very likely to say something meaningful by just doing babbling. In order to get the robot to actually learn some useful phonemes it is better to have the human to make the utterance and let the robot try to imitate. How well the robot will be able to repeat the same utterance depends on how well it has learnt the sensor-motor maps.

If the robot has mostly used babbling and had little or no previous interaction with its caregiver it is not likely to correctly map the sound of the caregiver when the caregiver uses his or her normal voice. In order to direct the robot to the correct utterance the caregiver may therefore need to adapt his or her own voice. This behavior can also be found in the interaction between a child and its parents and has been studied in (de Boer, 2005). When the robot answers with the intended utterance we give the robot positive feedback which causes the robot to store the current articulator positions in its cluster. This reinforcement was given through the keyboard in the current implementation, but more sophisticated methods could be used.

This step is only used insert new positions in the cluster and no training is going on in this step.

Experimental results

We performed three experiments using the architecture with babbling and imitation as described above. In the first experiment we test if the clustering algorithm is able to correctly group the positions it learns for 9 portuguese vowels. In the second experiment we use the learnt vowels and see how well the robot can recognize the same vowels when pronounced by different human speakers. Especially we look at the effect the visual features have on vowel recognition. In the third experiment we teach the robot some simple consonants and again look at the effect of using vision for recognition by studying the well know McGurk effect (McGurk and MacDonald, 1976) which can be expected when combining visual and auditory features.

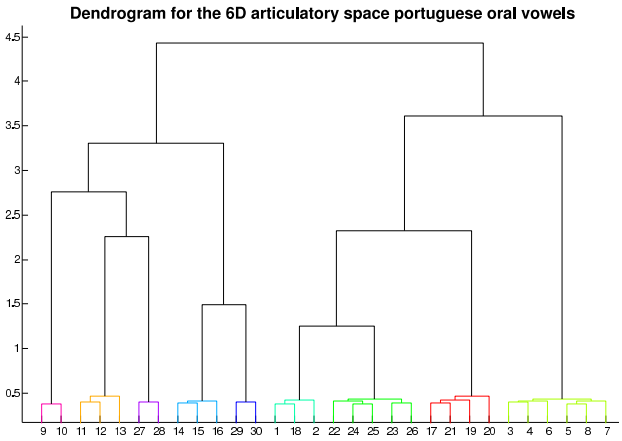


Figure 5: Dendrogram depicting the hierarchical clustering performed by the robot.

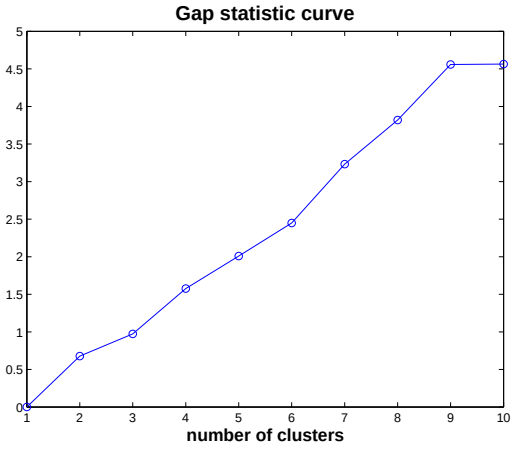


Figure 6: Gap statistic versus number of clusters. The growth of the curve stops at nine clusters.

Learning vowels

To create a sufficient number of valid training vowels for the robot, we created a dataset with 900 vowels, and then submitted them to the evaluation of 16 native speakers, so that they rejected or approved each vowel as a valid portuguese vowel and — for those that were approved — agreed or not in their phonological classification. From these 900 vowels, 281 were considered appropriated.

The original dataset was generated from nine prototype vowels in the 6D articulatory space, added with 10% of white noise.

Applying agglomerative hierarchical clustering to the present vowel dataset originated some good results, as we can see in figure 5. The nine vowel groupings depicted in different colors are clearly visible.

The dendrogram shown can be seen as a summary of the data structure that was detected by our simple dissimilarity measure: euclidean distance between 6D vectors and average dissimilarity between groups.

In ten performed trials, the Gap statistic consistently pointed to nine as the most natural number of clusters. One example of this result is presented in figure 6.

Vowel recognition

To be able to compare the results obtained in this work with the results obtained in (Hörnstein and Santos-Victor, 2007), we actually do not use the vowels positions learned by the cluster above, but instead use the positions learnt in the referred work. There the robot first learned its own sound-motor map by doing a completely random babbling. A caregiver then taught the robot nine Portuguese vowels by having the robot imitate the vowels and storing those that were successfully pronounced. As seen in Figure 7 the articulator positions used by the robot are similar to those used by a human speaker.

Next, the 14 speakers (seven males and seven females) were recorded while reading words that included the same nine Portuguese vowels. Each speaker read the words several times, and the vowels were hand labeled with a number 1 to 9. The amplitude of the sound was normalized and each vowel was then divided into 30 ms windows with 50% overlap. Each window was then treated as individual data which resulted in a training set of 2428 samples, and a test set of 1694 samples.

In addition to the original data we also extracted images from the video sequences that corresponded to each person pronouncing the vowels. Only one image for each person and vowel was extracted creating a training and test set of 63 images each. The images were then processed by the visual sensor in order to calculate the mouth opening in each image.

After the learning of the maps using random babbling the recognition rate for the human vowels in the test set were as low as 17.5%. We then used the data from the seven persons in the training set to imitate the robot's vowels to allow the robot to further train both the auditory-motor and the visual-motor maps. After the interaction with the persons in the training set, the recognition rate for the persons in the test set became 63.3%. If the robot was just presented with auditory input and was not allowed to see the person the recognition rate became 57.7%.

Learning consonants

We have also done some initial experiments with teaching the robot consonants using the methods described above. Each consonant is here modelled with a single target point in the articulatory space. It should be noted that the point by itself cannot reproduce the consonant. To reproduce the consonant we create a trajectory between two vowels that passes through the target point.

For this experiment the robot started with the three corner vowels [i], [a] and [u], and did an initial babbling by creating 1000 trajectories with 10 points along each trajectory.

In the second step we let the caregiver imitate the robot. We only created the straight trajectories [i] to [a], [a] to [u] and [u] to [a] as the alignment between the robot and the human utterances had to be made by hand, but these were sufficient to give the robot initial sensor-motor maps for the auditory and visual features of the caregiver.

The last step was to let the robot imitate the caregiver. We wanted to teach the robot three new phonemes /b/, /d/, and /g/ by having it imitating the utterances ba-ba, da-da, and ga-ga. This was done by feeding the last hearable sound before reaching the goal position of the consonant to the auditory sensor along with an image of the lip position at the goal position. The sound and the image were extracted automatically when the sound got below a threshold.

Teaching the robot a /b/ was pretty straight forward as it the robot could easily extract the main position from the visual feature. The latter two demanded a little more patience from the caregiver. The task got extra difficult since the synthesizer used does not create any clear consonants so we actually needed to inspect the resulting vocal tract position of the robot in order to decide if we were happy or not with the result. As we got happy with an utterance we stored the position in the motor cluster. The learnt positions can be seen in Figure 8.

Once the robot had learnt the positions we again switched roles and let the caregiver imitate the robot. After doing that the robot could easily recognise and reproduce the correct consonant. However we only did this experiment with a single caregiver so we do not expect the robot to generalize and correctly classify the same consonants when uttered by another speaker.

Finally we did a simple experiment were we tried to reproduce the McGurk effect by feeding the auditory sensor with ba-ba while feeding the visual sensor with ga-ga. Depending on the weight we put on the visual sensor relative to the auditory sensor the robot classify the utterance as either a ba-ba or da-da.

Conclusions

We have demonstrated how a humanoid robot can develop speech by using a combination of babbling and imitation.

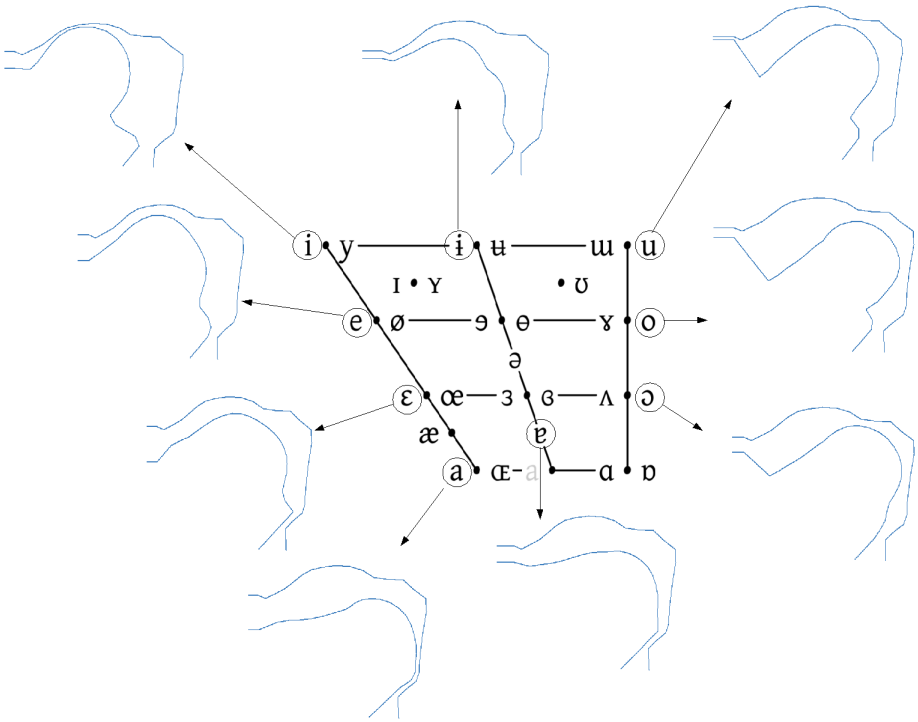


Figure 7: Articulator positions used by the robot for the Portuguese vowels. In the center we show the positions of the vowels in the International Phonetic Alphabet (IPA). The vertical axis in the IPA corresponds to the vertical position of the tongue and the horisontal axis to the front-back position when the vowel is pronounced by a human speaker. For the simulated articulator positions used by the robot the upper line corresponds to the soft palate and the lower line to the tongue. There is a good correlation between how the robot and a human articulate the vowels.

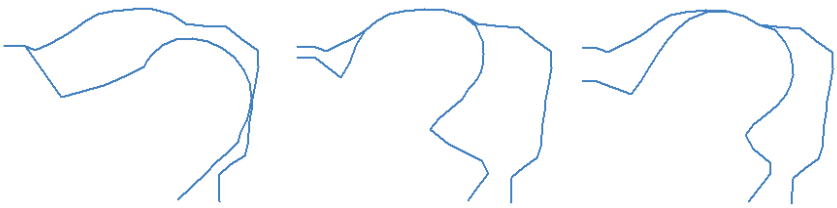


Figure 8: Learnt positions for the consonants /b/, /d/ and /g/.

While babbling make it possible for the robot to learn the map between its own sound and motor positions, interaction with a caregiver is more important for learning to map and understand human speech.

By letting the robot and the caregiver take turn in imitating each other it is possible both to teach the robot reproduce utterances made by the caregiver and learning which utterances that are useful for communication.

We have also shown that visual features can be helpful both to increase the recognizion rate of already learned phonemes and for learning new phonemes.

Acknowledgement

This work was partially supported by EU Project CONTACT and by the Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS Conhecimento Program that includes FEDER funds.

References

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, speech, and signal processing*, ASSP-28, no. 4.

de Boer, B. (2005). Infant directed speech and evolution of language. In *Evolutionary Prerequisites for Language, Oxford: Oxford University Press*, pages 100–121.

Fadiga, L., Craighero, L., Buccino, G., and Rizzolatti, G. (2002). Speech listening specifically modulates the excitability of tongue muscles: a tms study. *European Journal of Neuroscience*, Vol 15:399–402.

Fitzgibbon, A., Pilu, M., and Risher, R. B. (1999). Direct least square fitting of ellipses. *Tern Analysis and Machine Intelligence*, 21.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hastie, T. (2001). *The elements of statistical learning data mining inference and prediction*. Springer.

Hörnstein, J. and Santos-Victor, J. (2007). A unified approach to speech production and recognition based on articulatory motor representations. In *IROS07*, pages 3442–3447.

Kanda, H. and Ogata, T. (2007). Vocal imitation using physical vocal tract model. In *IROS07*, pages 1846–1851.

Kass, M., Witkin, A., and Terzopoulus, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*.

Liberman, A. and Mattingly, I. (1985). The motor theory of speech perception revisited. *Cognition*, 21:1–36.

Lien, J. J.-J., Kanade, T., Cohn, J., and Li, C.-C. (1999). Detection, tracking, and classification of action units in facial expression. *Journal of Robotics and Autonomous Systems*.

Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. *IEEE ICIP*, pages 900–903.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130.

Maeda, S. (1990). Compensatory articulation during speech: evidence from the analysis and synthesis of vocat-tract shapes using an articulatory model. In *Speech production and speech modelling, W. J. Hardcastle and A. Marchal, esd.*, pages 131–149. Boston: Kluwer Academic Publishers.

McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, (264):746–748.

Soares, C. and Bernadino, A. (2007). Mapping the vocal tract with a 2d vocalic articulatory space: applications to developmental robotics. In *Symposium on Language and Robots*.

Tian, Y., Kanade, K., and Cohn, J. (1999). Multi-state based facial feature tracking and detection. In *technical report, Robotics Institute, Carnegie Mellon University*.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2).

Viola, P. and Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE CVPR*.

Crossmodal Content Binding in Information-Processing Architectures

Henrik Jacobsson
henrikj@dfki.de
Nick Hawes
n.a.hawes@cs.bham.ac.uk
Geert-Jan Kruijff
gj@dfki.de
Jeremy Wyatt
j.l.wyatt@cs.bham.ac.uk

Abstract

Operating in a physical context, an intelligent robot faces two fundamental problems. First, it needs to combine information from its different sensors to form a representation of the environment that is more complete than any of its sensors on its own could provide. Second, it needs to combine high-level representations (such as those for planning and dialogue) with its sensory information, to ensure that the interpretations of these symbolic representations are grounded in the situated context. Previous approaches to this problem have used techniques such as (low-level) information fusion, ontological reasoning, and (high-level) concept learning. This paper presents a framework in which these, and other approaches, can be combined to form a shared representation of the current state of the robot in relation to its environment and other agents. Preliminary results from an implemented system are presented to illustrate how the framework supports behaviours commonly required of an intelligent robot.

Introduction

An information-processing architecture for robotics is typically composed of a large number of cooperating subsystems, such as natural language analysis and production, computer vision, motoric skills, and various deliberative processes such as symbolic planners. The challenge addressed in this paper is the production and maintenance of a model of the world for a robot situated in "everyday" scenarios involving human interaction. This requires a method for *binding* representations across the subsystems. This world model should adequately reflect the aspects of the world that are stable in the medium term, whilst incorporating more dynamic aspects.

Throughout this paper we will primarily consider a robot that can interact with a human and a set of objects on a tabletop. For example, when faced with a scene containing a red mug, a blue cup and a blue bowl, the robot may be asked to "put the blue things to the left of the red thing". For a system to be able to perform such a task effectively, it must be able to build a representation that connects the (low-level and modality specific) information about the world and the (high-level and amodal) representations that can be used to interpret the utterance, determine the desired world state, and plan behaviour. As resulting actions must be executed in the world, the representation must allow the robot to ultimately access the low-level (i.e. metric) information from which its higher-level representations are derived.

Any design for a system to tackle the above task must focus on creating such a representation, and grounding it in the environment of the robot. In addition to this, the engineering effort of integrating the various information-processing subsystems with the representation must be considered. After all, since the robot is an engineered system, every component must be put there by means of human effort.

The grounding problem is entangled with the engineering problem of subsystem integration and cannot be considered in isolation. Grounding can generally be seen as the process of establishing the relation between a representation in one domain with that of another. One special case is is when one of the domains is the external world, i.e. "reality":

> The term grounding [denotes] the processes by which an agent relates beliefs to external physical objects. Agents use grounding processes to construct models of, predict, and react to, their external environment. Language grounding refers to processes specialised for relating words and speech acts to a language user's environment via grounded beliefs. (Roy, 2005b) p. 8

In this paper we do actually not explicitly deal with reality. We will assume that, among the perceptual subsystem, some may have representations that more or less coincide with the aspects of reality of relevance for the tasks at hand. We also do not consider grounding of only linguistic symbols, nor to physical objects alone, and we do not assume that all representations must be grounded.

In the remainder of this paper we present a design for a subsystem of an information-processing architecture that is able to bind together subsystem-specific representations into a representation shared by the entire system. This binding system, henceforth the *binder*, tackles the problem of creating high-level shared representations that relate back to low-level subsystem-specific representations, as well as addressing the engineering issue of non-

[0]This work was supported by the EU FP6 IST Cognitive Systems Integrated Project "CoSy" FP6-004250-IP.

[0]This paper is an extended version of a paper submitted to HRI 2008.

intrusively integrating such representations into a real system.

The following section will discuss related approaches to similar problems, and identify requirements imposed by the task of creating a situated represention of the world. Following this, the information-processing architecture upon which the binder operates is presented. The binder will then be described, and this description will be followed by discussion about its properties, examples of its application and finally a concluding discussion relating to earlier work in the field.

Background and Motivation

Rather than attempt to address the complete spectrum of problems related to building and grounding representations, we will narrow the problem space by addressing the requirements for a *situated* representation of the *current state* that is able to support *deliberative processes* for a specific set of *scenarios*. These restrictions allow us to focus on particular properties of representations that are appropriate for our task domain, i.e. human-robot interaction linked to object manipulation (Hawes et al., 2007a) and human augmented mapping of an office environment (Zender et al.,). In particular, we are interested in binding together content from separate information-processing subsystems to provide symbols that can be used for deliberation and then action. By deliberation we mean processes that explicitly represent and reason about hypothetical world states, such as a possible interpretation of an utterance or a possible course of action. Such processes would be found in the deliberative layer of the CogAff schema (Sloman, 2001).

We can specify a number of requirements that for symbols. First, these symbols must be likely to be stable for the duration of the deliberative processes that they will be involved in. For example, a representation of an object from vision should remain stable across multiple image frames if that object is to be involved in a planning process. Second, these symbols must be represented at a level of abstraction appropriate for the processing they will be involved in. For example, a number of objects on a tabletop could be represented in a metric coordinate frame or as an abstracted symbolic representation. Each of these representations would be appropriate for reasoning at a particular level of detail. These two requirements are closely linked, so that the level of detail of a particular representation influences its temporal stability. These requirements have been directly informed by the requirements on representations for planning and acting in dynamic, uncertain worlds (Wood, 1993).

Furthermore, since these symbols must be produced by binding content across arbitrary information-processing subsystems that are operating on different data in parallel, it is unlikely that the binding of content can happen in a synchronous manner. Perceptual subsystems are typically event driven and to keep a representation of the current state as current as possible (especially in terms of real time constraints), it is important that perceptual information is processed as soon as it is generated. For example, in order to take into account and advantage of, perceptual priming in the system, then asynchronous crossmodal processing is essential. Incremental parsing of natural language is a good example of this. In (Brick and Scheutz, 2007), for example, the search for possible parses of an utterance was pruned through the context of the current scene. Therefore it is important that any representation of the current state can be incrementally and asynchronously extended as soon as data can be gathered and information can be processed (i.e. our binding process must be anytime).

Previous robotic systems which are able to bind information from one subsystem to another typically limit this kind of binding to linking linguistic references to objects created from vision. The first system that might even have conceivably encountered the problem was the Shakey system (Nilsson, 1984). This translated a constrained set of English language sentences into first order predicate calculus goal statements for the robot. Reference here was either non-specific (i.e. “move any box”), or non-ambiguous (each referent that needed to be specifically identified was given a unique name, e.g. “door D”). In making bindings of referents in the goal statement to the objects in the world the non-specific referents allowed lazy binding, so that binding was executed using a unification mechanism at plan execution time. This very late binding was only made feasible by the assumption of perceptual reliability, and by the other restrictions given above. However, later systems mostly follow Shakey in their choice of a parsimonious internal language that is essentially a direct mapping onto the qualities of objects that we express relatively straightforwardly in language, and which are naturally stable.

Current approaches, while following this choice of features on which to bind, attempt to bind referents from vision with language using a mixture of deterministic and probabilistic representations, and employing varying levels of abstraction. Mavridis and Roy (Mavridis and Roy, 2006) for example, have a single amodal world model, but one which contains linked deterministic continuous, stochastic qualitative, and stochastic continuous representations. They refer to these as being part of what they call a grounded situation model. In this case the linking is thus essentially not between pairs of properties in vision and language, but between all pairs properties of the same type (colour, position) by using a probability distribution over the bindings between. It is, at the time of writing, not yet fully implemented in a robot, and as far as specified makes no attempt to deal with the asynchronous change of representations in different parts of the system. In other systems (Roy and Mukherjee, 2005; Brick and Scheutz, 2007) binding can occur at a very early stage in processing, with information even from the speech signal influencing visual hypotheses as to the object being referenced, and vice-versa.

Engel and Pfleger (Engel and Pfleger, 2006) build more on the notion of gathering all data first, then generating a binding with the highest possible quality. For perceptual

Proc. LangRo'2007
978-972-96895-2-9

priming, this approach may not be very fruitful. We would however argue that it would be a grave mistake to not be able to utilise information from, and thereby build upon earlier work on symbol grounding. If, for example, the systems of (Roy, 2005b; Steels, 2006; Engel and Pfleger, 2006; Kruijff et al., 2006; Fransen et al.,), at least in principle, could be utilised in one and the same system, we would truly be able to take a step forward together as a community. Therefore *non-intrusiveness* is an important requirement on any binding system we build. In other words, it is important to make it straight-forward to integrate existing systems into any binding approach we take. This requirement also holds for the integration of existing perceptual and other subsystems. This is in part a requirement on the interfaces to a binding system, they must be kept simple and generic.

To summarise, the main requirements we have on our binder are:

- The symbols produced should be stable,
- they should have the appropriate level of abstraction (i.e. amodal *and* modal),
- they must be generated in an asynchronous, incremental, anytime manner,
- their production must be non-intrusive with respect to existing systems.

The Architecture

To demonstrate our approach to binding in practice, we have built an integrated robotic system to perform tasks in our tabletop HRI domain. The system has been presented in previous work (e.g. (Hawes et al., 2007a)), so we will only give a brief overview here. The design of the system is based on the CoSy Architecture Schema (CAS), a set of rules for designing architecture instantiations in a principled manner. The schema allows a collection of interconnected *subarchitectures* (SA), each containing a collection of processing components that can be connected to sensors and effectors. Each subarchitecture also contains a *working memory* (WM), which the components use to share information. Only components within a subarchitecture can write to the subarchitecture working memory, but all components can read from any working memory. We also allow for privileged components that can write to any working memory (thus supporting cross-architecture control mechanisms). The schema is implemented in our code using the CoSy Architecture Schema Toolkit (CAST), an open-source, multi-language implementation of CAS (Hawes et al., 2007b).

In our implementation we have subarchitectures for vision, communication, manipulation, planning, spatial reasoning, coordination and binding. Together they create a system that can learn and describe object properties in dialogue with a tutor, and also carry out manipulation commands that feature object descriptions based on the learnt visual properties. Each subarchitecture working memory contains specialised representations of the information processed by the attached components. For example, the visual working memory contains regions of interest generated by a segmentor and proto-objects generated by interpreting these regions, the communication subarchitecture contains logical forms generated from parsing utterances, and the spatial reasoning subarchitecture contains abstractions of physical objects with qualitative spatial relationships between them.

The Binder

Overview

The CAS-based architecture provides an ideal test case for the development of a situated representation. Each subarchitecture working memory contains specialised representations, and a subset of these could in principle contribute towards a general representation of the current state. In brief, our approach to tackling this problem has two parts: mapping from specific to general representations, and fusion of general representations. To enable specialised representations to contribute to the representation of the current state, each subarchitecture must provide a process that maps from the specialised representations into the more general ones (a process of abstraction). Each subarchitecture provides items of information to the binder (i.e. a separate binding subarchitecture) as *binding proxies*. A proxy is essentially a bundle of subarchitecture-specific (modal) information about, e.g., an object, a relationship, a collection of objects etc. The main constituent of a proxy is a set of attribute-value pairs, called *features* (such as colour, ontological category, connections to other proxies etc.). The proxies are essentially broadcast to the rest of the subarchitectures via the binder which attempts to *bind* proxies together based on whether their defining features agree on a common description. The structures that result from binding proxies are called *union*s, as they essentially contain the union of the features of the bound proxies. The set of unions represents the current best *architecture wide* hypothesis of the world state of the robot. This is based on the assumption that the underlying proxies and features are also the best hypotheses from the underlying subarchitectures.

The levels of abstraction of the binder and the other subarchitectures are conceptually illustrated in Figure 1. We will now describe in detail how the above is actually achieved.

Implementation

Our approach to generating a shared representation should not limit what SAs can express (based on the the non-intrusiveness requirement). Therefore the set of possible features is very broad:

Definition 1 A feature space $\Phi^x \in \mathbf{\Phi}$ is any data format in the space of all possible data formats, $\mathbf{\Phi}$. $\phi_i^x \in \Phi^x$ denotes an instantiation of a particular representation where x should be interpreted as any feature space name. □

For example, $\phi_{red}^{ColourLabel} \in \Phi^{Colour}$ denotes the colour "red" in the representation space of colours (the exact

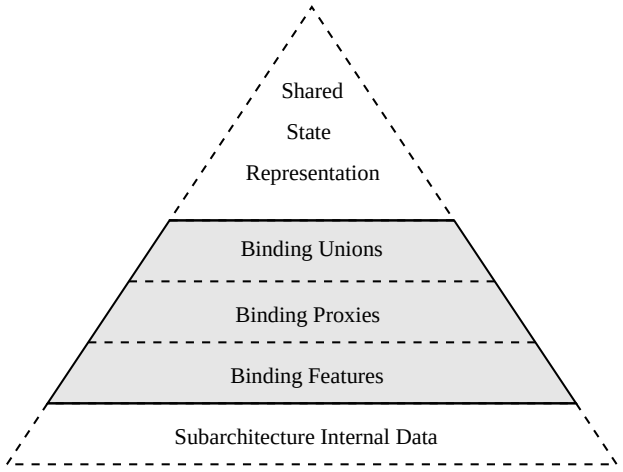


Figure 1: An illustration of how the binder mediates low-level and modality specific features from the SAs to form a common amodal representation of the world. The SAs are only involved in generation of features and proxies. Everything else is handled by the binder processes. SAs may take into account (or ignore) the common representation formed by the set of unified proxies. Conceptually, the top of the pyramid contains less, and more abstract, information than the base. The top level of the pyramid rests solidly on this base, since the features are always referred to from the unified abstracted data.

implementation of this representation is of no relevance here). In our CAST instantiation, $\mathbf{\Phi}$ corresponds to anything that can be represented by IDL-defined structs (including nested ones).

Information from the SAs is shared as a collection of proxies:

Definition 2 A binding proxy is a structure $p = \langle F_p, u_p \rangle$ where F_p is a set of instantiated features of different types (i.e. $F_p = \{\phi_1^{x_1}, \phi_2^{x_2} \dots \phi_n^{x_n}\}$) and u_p refers to a binding union with which the proxy is bound (see below). □

The unions should express information from proxies that, by all accounts (cf. Algorithm 1), refer to the same entity. Unions simply inherit the features of the bound proxies and are defined as:

Definition 3 A binding union is a structure $u = \langle F_u, \mathbf{P}_u \rangle$ where $\mathbf{P}_u$ refers to the subset of proxies unified by the union u and Fu is defined as the union of the features in all proxies in $\mathbf{P}_u$. □

The relationship between features, proxies and unions is illustrated in figure 2.

The problem for the binder is to assess whether two proxies are matching or not. By matching we mean that they should refer to the same thing. To do this, all new or updated proxies are compared to all unions on the basis of their respective features. The basis of this comparison is that each pair of feature types has an associated comparator function:

Definition 4 A feature comparator is a function $\Delta : \Phi^x \times \Phi^y \rightarrow \{true, false, indeterminate\}$ returning a value corresponding to whether a pair of feature instances are equivalent (or similar enough) or not. The comparator can also choose to not return a definite answer if the answer is undefined, or the uncertainty is too big (i.e. $indeterminate$). □

Obviously, $indeterminate$ is the only answer most such comparators can return, e.g. the comparison of a Φ^{Colour} and a $\Phi^{Position}$ is likely undefined[1]. For many pairs of features, however, there exist informative comparators. For example, features such as linguistic concepts can be compared to other concepts (with ontological reasoning (Engel and Pfleger, 2006)) or physical positions can be compared to areas.

Definition 5 A pair of feature spaces (Φ^x, Φ^y) is *comparable* iff $\exists(\phi_i^x, \phi_j^y) \in (\Phi^x, \Phi^y)$ such that $\Delta(\phi_i^x, \phi_j^y) \neq indeterminate$. □

The more pairs of features from different SAs that are comparable, the more likely it is that proxies from these SAs will be accurately matched.

To compare a proxy and a union, the corresponding feature sets are the basis for scoring:

Definition 6 The binding scorer is a function $S^+ : \mathcal{P} \times \mathcal{U} \rightarrow \mathbb{N}$ where $\mathcal{P}$ and $\mathcal{U}$ denote the set of all proxies and unions respectively and

$$S^+(p,u) = \sum_{\phi_i^x \in F_p} \sum_{\phi_j^y \in F_u} \begin{cases} 1 & \text{if } \Delta(\phi_i^x, \phi_j^y) = true \wedge \phi_i^x \neq \phi_j^y \\ 0 & \text{otherwise} \end{cases}$$

where F_p and F_u are the feature sets of p and u respectively. □

Note that identical features are not counted. This to prevent a union getting a higher score just because it is compared to one of its member proxies (this would sometimes prevent a proxy switching to a better union). The number of feature mismatches is also counted (i.e. with $true$ replaced with $false$ in S^+). That function is here denoted $S^- : \mathcal{P} \times \mathcal{U} \rightarrow \mathbb{N}$.

It is important to state that S^+ and S^- are implemented *asynchronously* with respect to the comparators. Until a comparator has returned an answer, S^+ and S^- will simply assume that the answer is neither $true$ or $false$, i.e. $indeterminate$.

S^+ and S^- are the basis for selecting the best among all unions for each new or updated proxy. This is conducted by the function `bestUnionsforProxy` described in Algorithm 1. The result of $best =$ `bestUnionsforProxy` is a set of zero, one or more unions. If $best = \emptyset$ then a new union will be created to for the proxy p alone (i.e. with all the features of p). If $|best| = 1$, then the proxy is bound to that union.

When $|best| > 2$ we are faced with a *disambiguation* problem. In those cases, we currently select a random union from $best$ for binding. The bindings are however

[1] Of course, in the implementation, such undefined comparators are never invoked. Mathematically, however, this is exactly what happens.

Proc. LangRo'2007
978-972-96895-2-9

```
bestUnionsforProxy(p, U)
Input: A proxy, p, and the set of all unions, U.
Output: Best union(s) with which a proxy should bind.
begin
    best := ∅;
    max := 0;
    for ∀u ∈ U do
        if S⁻(p,u) = 0 ∧ S⁺(p,u) > max then
            best := {u};
            max := S⁺(p,u);
        else if S⁻(p,u) = 0 ∧ S⁺(p,u) = max then
            best := best ∪ {u};
        end
    end
    return best;
end
```

Algorithm 1. The algorithm which computes the set of best candidate unions for being bound with a new or updated proxy.

sticky, i.e. if an already bound proxy subsequently matches a union in a larger "best"-list, then it will not switch to any of those unions. This to avoid excess processing in, and signalling from, the binder. This also helps to satisfy our requirement for symbols to be stable as far as possible. Disambiguation problems cannot be solved by the binder itself, but it can request help from others SAs in a general way. This may result, for example, in the communication SA initiating a clarification dialogue with a human tutor (cf. Section).

Relations and Groups

The proxies and unions described so far have been assumed to roughly correspond directly to physical objects. They may however correspond to arbitrary abstract entities as well. Two special proxies are implemented in a slightly different manner: groups of proxies and relationships between proxies.

Since proxies contain features that are of any representable type, proxies can also have features attributable to groups and relations, e.g. cardinality and relative metric information respectively and explicit references to relating proxies. Currently we handle groups in a fairly simple but effective way: a special kind of "group proxy" is created exactly like an ordinary binding proxy with all the features that the members of the group have in common (e.g. "the blue balls to the left of the mug" creates a group with features $\phi_{ball}^{Concept}$ and $\phi_{blue}^{ColourLabel}$ and with a spatial relation $\phi_{left\_of}^{SpatialRel}$-proxy to the $\phi_{mug}^{Concept}$-proxy. A separate process in the binding SA (the "group manager") then spawns of individual proxies which inherit the features of the group proxy. Every time an individual is bound to something, a new proxy is spawned[2]. To all the other processes, the individuals will appear as and endless supply of normal proxies.

The implementation of the relation proxies work in a similar way as for all other proxies. Spatial metric features, e.g. $\phi_{(x,y,z)}^{\mathbb{R}^3}$, could in principle be compared to a linguistic feature describing the same relation, e.g. $\phi_{left\_of}^{SpatialRel}$[3]. Since it has turned out that features that link relations to normal proxies and vice versa make the scoring inefficient, the relational structure between proxies are not fully incorporated in the scoring function. We are, however, currently reimplementing the scoring so that proxies with no comparable features still may be bound if they are part of matching relationships.

Examples

To illustrate how our binder supports a number of behaviours typically required of robots that interact with humans, the following sections present a number of examples taken from our implemented system.

Visual & Spatial Reference Resolution

Perhaps the most common use of information fusion systems is to interpret linguistic references in terms of visual information (cf. section). Our binder handles this task as an instance of a more general problem of information fusion. We will here consider the simple situation where we have a red object and two blue objects on the table. The objects are arranged in a straight line of alternating colours. The human then asks the robot to "put the blue objects to the left of the red objects".

We will start our example in the visual subarchitecture, where change detection, tracking and segmentation components create representations of the objects in the scene. These objects have 3D poses and bounding boxes and a number of slots for visual properties such as colour, shape and size. These slots are filled by a recogniser that has been previously trained (see Section) using input from a human trainer (Skočaj et al., 2007). In our example we assume the recogniser correctly extracts the colours of the objects as red and blue. When the scene becomes stable (determined by the change detector) the visual subarchitecture binding monitor creates a proxy for each of the currently visible objects. As the visual property recogniser processes the objects, the monitor updates the proxies with features reflecting these properties. This is an incremental process, so the visual proxies are updated asynchronously as the objects are processed. At this point only the visual proxies are present in the binding working memory, each one is bound to its own union.

The presence of objects in the visual working memory is also noticed by the components in the spatial subarchitecture. These abstract the objects as points on the tabletop, and the spatial binding monitor creates a proxy for each. These proxies are tagged with the ID of the visual proxy for the corresponding object so they are bound correctly[4]. Concurrently with the proxy creation, qualitative spatial relations between the spatial objects are added to

[2] With some obvious limitations to prevent infinitely many proxies to be generated when members of different groups merge. Also, the number of elements in a group can be specified and used to limit the number of individuals spawned off.

[3] Of course with respect to a reference position proxy, e.g. the position of the robot's camera. The relations can be N-ary.

[4] A similar, but more general, functionality could be generated by matching location-derived features.

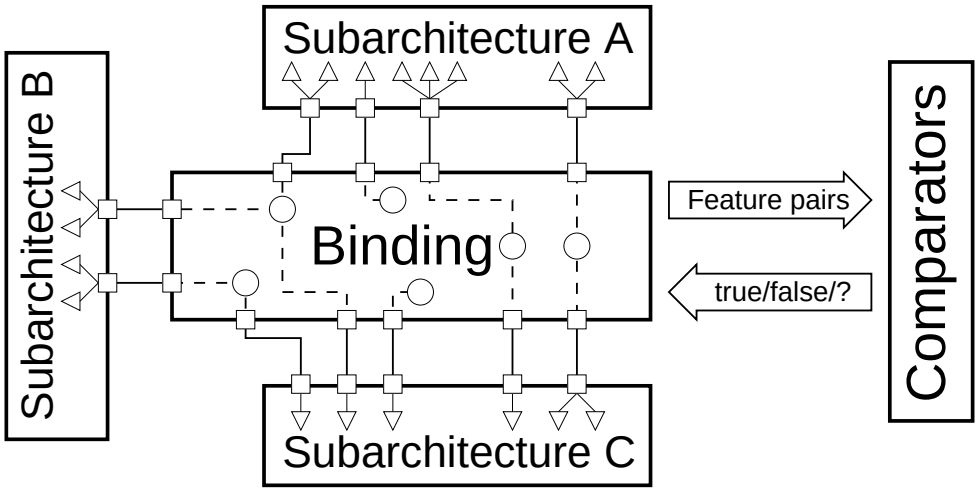


Figure 2: The relationship between binding features (triangles), proxies (squares) and unions (circles) in the context of the binder, subarchitectures and comparators. For clarity, the comparators are shown as a separate box here. They are, however, typically residing withing the subarchitectures themelves. The proxies represent concepts, objects, actions and relations between proxies (relations not shown here, but these are also mirrored in associated unions). Through the proxies, via the unions, all the features of the involved subarchitectures can be accessed across the system. This is the key to the two level representation, definitions are accessed via the proxies, shared via the unions, and described in an arbitrary level of detail in the features.

working memory. These are generated by components using potential-field-based models of spatial relations (Brenner et al., 2007). In our example the two blue objects are to the left and to the right of the red object respectively. They are both also near the red object (but not near each other). As these relations are added, the spatial binding monitor reflects them on the binding working memory as relation proxies between the spatial proxies. The binder uses these as the basis of relations between the unions featuring the spatial proxies. This provides our basic model of the current state.

When the human speaks to the robot, a speech recognition module in the communication subarchitecture is triggered. The resulting speech string is written to the communication working memory. This triggers a cycle of deep sentence analysis and dialogue interpretation, yielding a structured logical description of the utterance's content. From this structure the communication binding monitor generates communication proxies for the discourse referents and the relations between them. These proxies include features that can match against both those attached to visual proxies (colour, shape and size), and those attached to spatial proxies (relations based on spatial preposition). In the example two proxies are generated: one normal proxy for the red object, and one group proxy for the blue objects. The binder uses the features of these communication proxies to bind them into unions with the visual and spatial proxies. In the example the $\phi_{red}^{ColourLabel}$-proxy is bound together with the visual and spatial proxies relating the red object, and the $\phi_{blue}^{ColourLabel}$-proxies (spawned from the corresponding group proxy, see section) bound with the remaining proxies for the blue objects. This provides the system with an interpretation of the utterance in terms of the visual scene.

For our system the process of reference resolution involves simply ensuring that the communication proxies referring to visual entities (i.e. those referring to objects in the tabletop scenario) are bound to unions that have a visual component. If the utterance contains spatial language, then relation proxies are generated by the communication binding monitor. This causes the binding process to bind proxies via the relations between proxies as well as the features of single proxies. Failure to bind proxies can trigger a number of different processes, as described in .

Planning and Execution

Once the system has successfully interpreted the utterance in the previous example, it must now generate some behaviour. In addition to creating proxies, the interpretation of the utterance also produces information about the purpose of the utterance. In this case the utterance is determined to be an instruction, and this causes the mediation subarchitecture to generate a motive to act on the instruction. This process involves re-interpreting the utterance as a planning goal in MAPL, the multi-agent planning language used by the system. This interpretation is carried out by a general-purpose process that maps between verbs and planning operators (Brenner et al., 2007). Once the system has this goal, the planning subarchitecture is used to generate and execute a plan to satisfy it.

To create a plan, the planning subarchitecture needs to generate a description of the initial state for the planner to operate on. This is done by translating directly from the binding unions (accessed through the communication proxies stemming from the utterance) and their features into objects and fact descriptions in MAPL. Once a plan has been created, the execution components in the plan-

Proc. LangRo'2007
978-972-96895-2-9

ning subarchitecture start working through the plan triggering execution steps followed by execution monitoring steps.

In our current system we only have actions related to manipulation (pick up and put down), so all plan actions are forwarded to the manipulation subarchitecture. As the planning process used binding unions as input, the plan actions are also expressed in terms of these unions. The manipulation subarchitecture cannot operate on the symbolic representation used by the planner, but requires the detailed metric information generated by processes in the visual subarchitecture to support grasping. By following links from the unions in the planning action via the visual proxy to the object information in the visual subarchitecture, the manipulation processes get access to the necessary metric data. As the processes in the visual subarchitecture run constantly, this metric information is always kept consistent with the world. The binding structures, however, remain stable across these changes (unless they are significant enough to alter the spatial relations). Support for access to low-level information via high-level symbols, and support for concurrent updates to this low-level information whilst keeping the symbols stable, demonstrate two of the benefits of our two-level approach to represent the current state of the world.

In our example, the plan involves a single pair of pick and put actions. The blue object on the right of the scene is moved to the left of the red object. The other object is already to the left of it so it is not moved. After each object is moved the planning subarchitecture triggers an execution monitoring step. This step involves creating a new representation of the current state from the unions on binding working memory, and comparing them to state predicted by the plan. For the monitoring step in the example to complete successfully the unions must reflect that the moved blue object is now at a position to the left of the red object. Being able to monitor for such abstractly-specified conditions demonstrates the benefit of generating symbolic states on demand from a dynamically updated representation of the current state.

An Example of Interactive Learning

Our example system learns the visual properties of objects through dialogue with a human tutor (an interaction that can take many forms (Thomaz, 2006)). The tutor trains the system with sentences such as "this is a red thing" and "this is a small blue thing". The visual properties are ultimately learnt by a continuous learning system (Skočaj et al., 2007). Our approach to binding naturally supports the creation of training examples for this learning system. When an object is placed in front of the robot, the visual subarchitecture processes the object as described previously, ultimately creating a visual proxy for it. When the tutor makes an assertion about the object, we use recency information to bind the communication proxy for the deictic reference to the newest visual proxy[5]. The communication proxy contains binding features for all of the adjectives used in the utterance. When the visual subarchitecture binding monitor is informed its proxy has been bound into a union (via CAST change events), it inspects the union to see what features are present that it didn't add itself. These features represent information about the object from other modalities that the visual (learning) subarchitecture can choose to learn. Currently, we take a fixed subset of features present in the union and use them to generate input to for our learner. In theory these restrictions could be removed and features provided by other modalities could be used by any subarchitecture to learn cross-modal information. This simple way of driving cross-modal learning systems demonstrates a benefit of

[5]In this instance we are using recency as a substitute for a more complex process of reference resolution.

Generation of Clarification Events

It is not always possible for the binder to find unique bindings for proxies. For example, consider a scene with two red objects on the table which causes two visual ϕ_{red}^{Colour}-proxies to be created and bound into separate unions. The human then asks the robot to "pick up the red thing", creating a $\phi_{red}^{ColourLabel}$-proxy from the discourse. In this situation `bestUnionsforProxy` will return a set containing the two visually red unions. In the near future we plan to use situations such as this as general purpose triggers for generating *clarification behaviour*. For example, consider the case where the visual proxies (and thus their unions) have some mismatching features that separate them (e.g. ϕ_{round}^{Shape} and ϕ_{square}^{Shape}). In this case, this could lead directly to the system generating a goal to determine if the object being referred to by the human has one of the mismatching features. In this example the robot could ask a question about the distinguishing feature (e.g. "do you mean the *round* red thing?"). In principle, however, the general purpose nature of the binding system means that any subarchitecture that can provide a particular binding feature could satisfy such a request for information (i.e. not only dialogue).

The situation where the ambiguous unions have matching feature sets raises a different type of clarification problem. Rather than generating a need for a particular type of feature information to clarify a binding, resolving this situation requires a direct reference to the target object to allow binding. For example, the robot may have to ask "which red thing do you mean?", "do you mean this one?" (whilst pointing), or "do you mean the one on the left?". Alternatively the binding system could draw on information from other modalities to determine things such as the likelihood of the object being involved in a pick-up action (e.g. a feature $\phi_{true}^{Reachable}$), or saliency of the object given the human's perspective on the scene. Building support for the sharing of such information via proxies allows a more general notion of saliency and attention to be built into the system.

Discussion

In the following sections we discuss the properties of the representation and binding system presented previously,

including how it relates to our original requirements.

Modal and Amodal Representations

As is apparent in the planning example above (Section), from the point of view of a particular SA, a union is an *amodal* entity. But despite this, it also contains a set of *modal* properties of which some have semantics for particular SAs. The binder mixes amodal and modal representations such that modality-independent proxies and unions can be used for symbolic processing while at the same time they contain references to modal representations, i.e., via the features. Moreover, if a feature space is used which supports the ability to refer to data *inside* local SA WMs, data types that have not been declared as features can still be shared with all other SAs if required.

Lazy Binding and Locally Stable Symbols

In our binder, the only thing an SA typically needs to keep track of, once a binding proxy is created, is the proxy itself. Once created as a candidate for becoming part of a union, the proxy indeed *acts* as a proxy into any other information from its union. This means that any SA-internal symbols can be made isomorphic to the indexes of the proxies and the SA does not need to necessarily take into account whether the proxy is in a union or not.

This can be a very powerful simplification for many types of SAs. For example, consider the navigation dialogue scenario where the user tells the robot to "go to the kitchen" (Zender et al.,). Now, if the robot has yet to discover the kitchen, the discourse referent proxy containing $\phi^{Concept}_{kitchen}$ can not really be bound to any kitchen in the map. As soon as some other process identifies and defines a kitchen in the map, the utterance's $\phi^{Concept}_{kitchen}$-proxy can be bound. Whether or not this binding takes place, the $\phi^{Concept}_{kitchen}$-proxy remains intact and can still be referred to internally the same way independently of if whether it is bound or not.

Scalability

The theoretical properties of the binder are irrelevant if it cannot be implemented in an effective way. Potentially, the binder may become a bottleneck in the architecture since it may receive features and proxies from all involved subarchitectures[6]. To overcome this, we have implemented the binder as several smaller components, each responsible for basic tasks (e.g. one component for invoking and collecting the results of comparators, one that generates the unions based on the scores etc.). All these components can be replicated and put on different physical nodes, sharing the computational load. Moreover, the feature comparisons are made externally to the binder and thus the computational load is further distributed[7].

The problem of making the binder scalable is in part addressed by the role of abstraction in the system. The data "closest" to the binder (i.e. the unions) are abstracted from the much more abundant features (which are primarily processed by the SAs themselves). This means the binder only has to operate on an abstracted subset of all of the information in the system.

Incremental Asynchronous Binding

As mentioned in Section it is desirable to do both early and anytime binding. We achieve this in our implementation of the binder by allowing all components operate on the data asynchronously. This makes the binding quite efficient since any processing tasks (feature comparisons, scorings, union creation etc), will also be carried as soon as is possible.

For example, in the in Section , the visual and spatial proxies are initially bound as basic object abstractions. Following this, SA-specific components gradually add more information about the objects (visual properties and spatial relations), which cause the proxies, and their unions, to be asynchronously updated with features and relations.

The anytime properties of the binder also mean that any comparison that is finished early may help in forming unions before any additional comparisons are made. Thus, hypothetical unions may be formed in an early stage and then refuted if conflicting information later comes in. This may of course cause an overhead in that incorrect unions are temporarily created, but for time critical SAs this may be not be critical. SAs with higher quality demands can always wait until the bindings have "settled".

Demands on Subarchitectures

From an engineering point of view, subarchitecture designers have to provide a number of things to support the binding process: 1. the feature definitions (if not already existing), 2. a binding *monitor* component that analyses local SA WM content and generates and mediates proxies onto the binding working memory, and 3. the comparators.

The comparators can be based upon any kind of computational process from simple equivalence testing like string matching to ontological, spatial or visual reasoning etc. The comparators may also be learnt models and can even be learnt online, on data extracted from unions, while the binder is operating (as presented in Section). Moreover, a comparator may be *context sensitive*, i.e. it can take into account all other information on the binding WM to make its assessment (cf. (Roy, 2005a)). It is also possible that the comparator itself triggers the SA to generate more features to complete a proxy's description. There are many possibilities since few limitations are imposed by the design of the binder.

The integration with the binder is fairly non-intrusive in the sense that none of the things that need to be provided should have any implications on any other part of the SA. Also, the SA only minimally needs to write features and proxies, and does not have to process what is happening on the binder at all (apart from the comparators that need to be reactive, but in some cases an SA may simply use

[6] This is one reason why SAs should be conservative about generating proxies.

[7] In the implementation, however, some trivial comparisons are actually handled internally in the binder.

Proc. LangRo'2007
978-972-96895-2-9

features for which comparators are already implemented). A slightly deeper integration problem occurs when an SA needs to utilise the contextual information represented by the unions (e.g. for priming, such as for incremental parsing). This is, however, arguably non-intrusive as well, as all features that are unknown to the SA in question can safely be ignored (cf. section).

The feature set space is also highly open-ended. An added feature definition will not affect, and will depend on, the earlier features in any way. Every subarchitecture can and will only deal with the features it knows about. This means that it is a fairly low cost to add features into the system.

One problem for some designers may be that expressive models of beliefs (e.g. Bayesian) are being robbed of their expressiveness when the comparator can only return three values. This is however a better situation than the opposite. There is also nothing that prevents a comparator to reason about degrees of belief in the setting of the context up to the point of the final decision.

In order for the binder to perform well, the designers need to be conservative. For example, proxies should not be generated excessively (for example, just to see if they will be bound or not) since it may disturb other SAs. And new feature types should primarily only be introduced if also a comparator for this feature can be defined.

If conservatism is not employed, the binder will not perform well. There are other traps too. Since features and comparators are representable in very open-ended formats, the SA designer has very few limitations in what can be done. This is of course an advantage in many cases. But many creative interpretations of "features", "proxies" and "comparators" will simply not yield desired results. For example, conflicting features can be inserted into a proxy, but that violates the proxy-as-best-hypothesis assumption.

Another problem is if the SAs can only provide features that are SA specific and incomparable with most features from other SAs. In such cases the binder would not be able to form any unions. It is thus important to have comparable features in mind when integrating SAs into an architecture.

Conclusion

In this paper we have presented a method for generating a stable, yet asynchronously updated, model of the current state of the world for a situated information-processing system such as an intelligent robot. The amodal model emerges from the incremental fusion of information abstracted from model data, and satisfies the requirements we specified for such a system. Although our system has been fully implemented, we have yet to run an experimental evaluation on it. In place of this we illustrated our system with a number of examples from the scenarios we are tackling.

References

Brenner, M., Hawes, N., Kelleher, J., and Wyatt, J. (2007). Mediating between qualitative and quantitative representations for task-orientated human-robot interaction. In *Proc. IJCAI '07*.

Brick, T. and Scheutz, M. (2007). Incremental natural language processing for hri. In *Proc HRI '07*, pages 263–270.

Engel, R. and Pfleger, N. (2006). Modality fusion. In Wahlster, W., editor, *SmartKom - Foundations of Multimodal Dialogue Systems*, Cognitive Technologies, pages 223–235. Springer.

Fransen, B., Morariu, V., Martinson, E., Blisard, S., Marge, M., Thomas, S., Schultz, A., and Perzanowski, D. Using vision, acoustics, and natural language for disambiguation.

Hawes, N., Sloman, A., Wyatt, J., Zillich, M., Jacobsson, H., Kruijff, G., Brenner, M., Berginc, G., and Skocaj, D. (2007a). Towards an integrated robot with multiple cognitive functions. In *Proc. AAAI '07*.

Hawes, N., Zillich, M., and Wyatt, J. (2007b). BALT & CAST: Middleware for cognitive robotics. In *Proc. IEEE RO-MAN 2007*, pages 998 – 1003.

Kruijff, G.-J., Kelleher, J., and Hawes, N. (2006). Information fusion for visual reference resolution in dynamic situated dialogue. In Andre, E., Dybkjaer, L., Minker, W., Neumann, H., and Weber, M., editors, *Proc. PIT '06*, pages 117 – 128.

Mavridis, N. and Roy, D. (2006). Grounded situation models for robots: Where words and percepts meet. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2006*, pages 4690–4697. IEEE/RSJ.

Nilsson, N. J. (1984). Shakey the robot. Technical Report 323, AI Center, SRI International.

Roy, D. (2005a). Grounding words in perception and action: Insights from computational models. *Trends in Cognitive Science*, 9(8):389–96.

Roy, D. (2005b). Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205.

Roy, D. and Mukherjee, N. (2005). Towards situated speech understanding: visual context priming of language models. *Computer Speech & Language*, 19(2):227–248.

Skočaj, D., Berginc, G., Ridge, B., Štimec, A., Jogan, M., Vanek, O., Leonardis, A., Hutter, M., and Hawes, N. (2007). A system for continuous learning of visual concepts. In *International Conference on Computer Vision Systems ICVS 2007*, Bielefeld, Germany.

Sloman, A. (2001). Varieties of affect and the CogAff architecture schema. In *Proc. the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, pages 1–10.

Steels, L. (2006). Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21(3):32–38.

Thomaz, A. L. (2006). *Socially Guided Machine Learning*. PhD thesis, Massachusetts Institute of Technology.

Wood, S. (1993). *Planning and Decision Making in Dynamic Domains*. Ellis Horwood.

Zender, H., Jensfelt, P., Óscar Martínez Mozos, Kruijff, G.-J. M., and Burgard, W. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proc. AAAI '07*.

Interactive Learning and Cross-Modal Binding - A Combined Approach*

Henrik Jacobsson[1], Nick Hawes[2], Danijel Skočaj[3], Geert-Jan M. Kruijff[1]
[1] Language Technology Lab, DFKI GmbH, Germany, [2] School of Computer Science, University of Birmingham, UK
[3] University of Ljubljana, Slovenia,

Introduction

To function properly in the world, a cognitive system should possess the ability to learn and adapt in a continuous, open-ended, life-long fashion. This learning is inherently cross-modal; the system should use all of its percepts and capabilities to sense and understand the environment, and update the current knowledge accordingly. For the life-long learning to be effective, it is also important to be able to incorporate knowledge from other knowledgable cognitive systems through *interactive learning*. For this to be "socially acceptable", it is important to support a wide variety of tutoring channels. For example, to treat the tutor only as a source for linguistic labels is not a natural way of communication and is thus not very effective from the human's point of view. For an excellent and deep account of proper design considerations for socially interactive learning systems see (Thomaz, 2006)

A prerequisite for interactive learning is the successful interpretation of the meaning of references used in dialogue with a human. The robot must therefore be able to form associations between information in different modalities, e.g., between linguistic references and visual input (Roy, 2005). Forming these associations is a process we refer to as *cross-modal binding*. We are developing a multifaceted approach to binding, and in this extended abstract we address the virtue of the symbiosis of binding and interactive learning.

Cross-Modal Binding

We treat the binding of linguistic and visual content as an instance of a broader cross-modal binding problem: to enable a broad and open-ended set of modalities to contribute towards a common representation of abstract concepts, objects, and actions, and N-ary relations between them. For example, for a robot to successfully determine the correct response to the "give me the blue mug that's to the right of the plate" it must be able to correctly interpret the references to the objects, the action, and the spatial relationship.

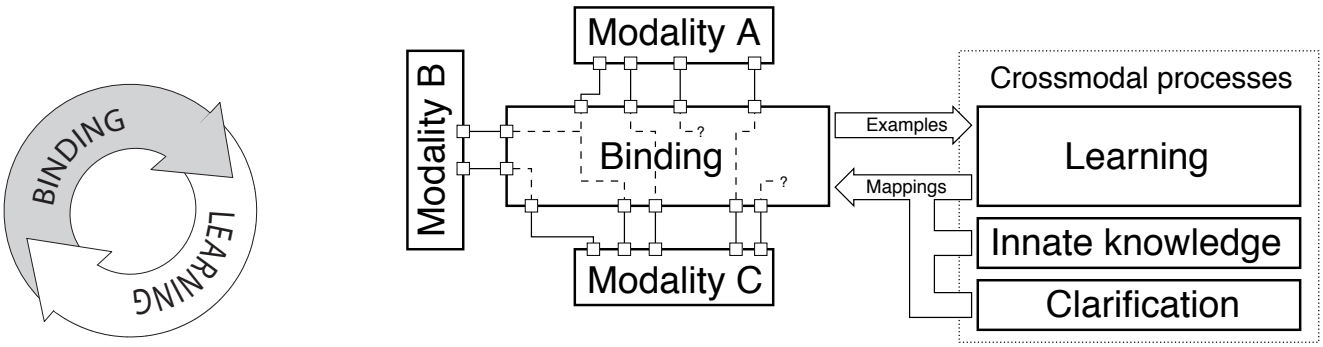
Typical robotic systems are composed of specialised subsystems, e.g. vision, manipulation, dialogue, reasoning etc. For N subsystems there are $N(N-1)/2$ potential interfaces between them. Building associations in this manner can quickly become expensive to manage both at design- and run-time. To avoid this, we employ a two-level approach to binding. The bottom level corresponds to subsystem specific representations. The second level represents objects, actions and relations by bundling together sets of *features* abstracted from the first level representations. These "bundles" represent a subsystem's best hypotheses about the objects, actions and relations in its modality. To build a common representation from all its subsystems, a number of *binding processes* then operate on this more abstract level of information. This is illustrated in Figure 1. Further information is available in previous work (Hawes et al., 2007). The focus of this abstract is that the information used to associate features across modalities may be learned, and that this two-level system naturally supports such learning.

Cross-Modal Learning

When the binding processes establish associations between bundles of abstracted features, these associations implicitly link features from these modalities. Some of these links will represent known cross-modal mappings between features, but others may represent valid mappings that the system does not know about. For example, in the utterance "give me the blue mug that's to the right of the plate" visual colour features (blue pixels) may be implicitly linked to linguistic colour features ("blue") via an association formed from a type description ("the mug"). When the binding of the object descriptions succeed, the binder can generate novel training examples for a learning module. In the case above, the binder would generate the training examples for updating the representations of "blue", "the mug", "to the right of", and "the plate". In this way, the system can increase its current knowledge without being explicitly instructed, and without training examples being provided separately. An idealised learner would try to use all the inferred information and data from all modalities to *co-train* (cf. Levin et al., 2003) its representations in other modalities as well.

Any learning method using binding processes for training will be thus fed by a stream of examples of cross-modal associations. The open-ended nature of this input makes it important that any learning systems used are *incremental*; the learning process should continue to im-

*This work was supported by the EU FP6 IST Cognitive Systems Integrated Project "CoSy" FP6-004250-IP.



(a) (b)

Figure 1: Cross-modal binding and incremental learning are tightly integrated (a) and feed each other information regarding mappings and learning examples respectively. They are also integrated in the context of the modalities themselves as well as other cross-modal processes that can assist the binder prior to, and in parallel with, learning (b). Clarification is one way of explicitly establishing mappings, e.g. through dialogue. Over time, the dual interaction between these subsystems results in increasingly informative learning examples and mapping suggestions.

prove the learned models from the incoming examples. The learned models should also be capable of determining their own level of confidence so that they contribute to the binder only when the representations are sufficiently stable and confident.

Since the learners can potentially know what kind of examples will help them train, they may also trigger explicit clarification behaviour. This will allow more focused clarification behaviour than the binder could trigger on its own (for ambiguous bindings etc.). For example, if a learned colour classifier is uncertain about labelling a visual object as “orange” or “red”, it could trigger a clarification behaviour to distinguish between *precisely* these labels. Or it could even issue a command to the manipulation module to turn the object in order to provide additional visual information to resolve ambiguity.

Taken together, the learning, binding and clarification behaviours of the system form a strong basis for a wide range of tutor-robot interactions. There is potential for mixed initiative dialogue since the agent will autonomously ask for information. Despite this, the learning is essentially autonomous and can passively listen for examples in case the agent is engaged in other behaviours. Additionally, by using abstracted representations of subsystem-specific representations, many of the mechanisms for reasoning and learning can be reused across very different domains. For example, general-purpose clarification planning mechanisms can be employed across different modalities.

Conclusion

In our approach, the binding processes and learning processes thus form a symbiosis, where they both benefit from the information they feed each other. The learners benefit from helping the binder to make cross-modal associations since it will then be able to make more (and potentially better) bindings, potentially resulting in more training data for the learners. By supporting the learning of cross-modal associations, e.g. between colour labels and visual colour representations, the binder may implicitly be able to associate information from other feature spaces, e.g. about spatial relations, via these learnt assoications. Moreover, an intelligent robot employing this approach also benefits from the symbiosis; its primary source of cross-modal mapping information (information required to support linguistic interaction) can be shifted from costly, deliberate tutoring to incrementally learned associations.

References

Hawes, N., Sloman, A., Wyatt, J., Zillich, M., Jacobsson, H., Kruijff, G., Brenner, M., Berginc, G., and Skočaj, D. (2007). Towards an integrated robot with multiple cognitive functions. In *Proc. AAAI ’07*.

Levin, A., Viola, P. A., and Freund, Y. (2003). Unsupervised improvement of visual detectors using co-training. In *ICCV*, pages 626–633, Nice, France.

Roy, D. (2005). Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205.

Thomaz, A. L. (2006). *Socially Guided Machine Learning*. PhD thesis, Massachusetts Institute of Technology.

Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction

Geert-Jan M. Kruijff[1], Pierre Lison[1,2], Trevor Benjamin[1,2],
Henrik Jacobsson[1], and Nick Hawes[3]

[1]Language Technology Lab, German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken Germany

[2]Dept. of Computational Linguistics, Saarland University, Saarbrücken Germany

[3]Dept. of Computer Science, University of Birmingham, Birmingham United Kingdom

gj@dfki.de

Abstract

The paper presents work in progress on an implemented model of situated dialogue processing. The underlying assumption is that to understand situated dialogue, communicated meaning needs to be related to the situation(s) it refers to. The model couples incremental processing to a notion of bidirectional connectivity, inspired by how humans process visually situated language. Analyzing an utterance in a "word-by-word, left-to-right" fashion, a representation of possible utterance interpretations is gradually built up. In a top-down fashion, the model tries to ground these interpretations in situation awareness, through which they can prime what is focused on in a situation. In a bottom-up fashion, the (im)possibility to ground certain interpretations primes how the analysis of the utterance further unfolds. The paper discusses the implementation of the model in a distributed, cognitive architecture for human-robot interaction, and presents an evaluation on a test suite. The evaluation quantifies the effects linguistic interpretation has on priming utterance processing, and discusses how the evaluation can be extended to include situation context.

Introduction

The environments in which we deploy our robots provide them with rich, perceptual experiences. And language provides a combinatoric system that enables us to talk about those environments in a rich variety of ways. The problem is of course then how we can figure out, what an utterance really is supposed to mean in a given context. From psycholinguistics and cognitive science we know that humans use context information to do this. Humans do not wait with processing an utterance until they have heard the end of it. On the contrary. While processing an utterance, they link unfolding interpretations to the dialogue- and situated-context to filter out unlikely interpretations. They use their understanding of the situational context to disambiguate and refine how they comprehend an utterance, and at the same time use what is being talked about to selectively refine their situation awareness. Using context, they pick those meanings out of the myriad of possible meanings, to focus on just those that seem most appropriate in the given context (Altmann and Steedman, 1988; Altmann and Kamide, 2004; Knoeferle and Crocker, 2006).

In this paper, we discuss an implemented model that enables a robot to understand situated dialogue in a similar way. The model relies on explicitly grounding dialogue in the situated context. The main idea is to use an *incremental* model for dialogue analysis, and connect step-by-step the unfolding possible *linguistic* interpretations of an utterance to information about the visually situated context. From this interconnection we can then derive what visual objects are being talked about, and whether the way these referents are referred to, and put into relation, can be grounded in the situated context. We use insights from psycholinguistics in postulating what factors in the visually situated context might play a role (Altmann and Steedman, 1988; Altmann and Kamide, 2004; Knoeferle and Crocker, 2006), and how they affect priming of utterance processing.

Our approach is related to other recent work on incremental language processing for dialogue systems (Allen et al., 1996; Mori et al., 2001; Rosé et al., 2002), and for human-robot interaction (Brick and Scheutz, 2007) (B&S). Like B&S we analyze an utterance for its meaning, not just for syntactic structure (Allen et al., 1996; Mori et al., 2001; Rosé et al., 2002). We make several advances, though. The model incrementally analyzes utterance meaning not only at the grammatical level, but also at dialogue level. B&S only consider the former (parsing). By interpreting an utterance also relative to the dialogue context, the model allows different levels of linguistic description to constrain possible interpretations (Altmann and Steedman, 1988; Stone and Doran, 1997). This presents several advantages. We can (linguistically) resolve contextual references such as deictic pronouns and anaphora. This resolution relates references made in the current utterance to ones made already earlier in the dialogue – i.e., ultimately to visual objects that have already been identified. Furthermore, we can use the dialogue "move" of the utterance to determine what *needs* to be bound. For example, in a greeting like "Hi there!" the model does not need to try and bind "there" to a location.

A further advance is that we adopt a "packed" representation of the linguistic interpretations (Oepen and Carroll, 2000; Carroll and Oepen, 2005) to efficiently handle alternative (i.e. ambiguous) meanings. Any grammar of a reasonable size will generate multiple syntactic-semantic analyses of an utterance. This can easily result in hundreds of alternative analyses that would need to be checked. A packed representation represents all the in-

formation shared across alternative analyses only *once*, which greatly reduces the amount of linguistic content we need to ground. These packed representations are subsequently related to information about the situation and ongoing tasks (Allen et al., 2001; DeVault and Stone, 2003; Gorniak and Roy, 2007). This essentially comes down to trying to resolve how a meaning refers to the current context (Stone and Doran, 1997; Brick and Scheutz, 2007). Intuitively, if a meaning presents an unresolvable reference, or an unresolvable assertion about spatial organization, then it can be discarded.

An overview of the paper is as follows. We start by providing a brief overview of insights of how humans process situated utterances, and position our approach to other work in AI and HRI. We then present our approach. We discuss its implementation using the CoSy Architecture Schema toolkit (Hawes et al., 2007a; Hawes et al., 2007b). Using a test suite with a variety of visual scenes, we evaluate our approach in a systematic way on different types of potential linguistic ambiguity. We measure the effects of linguistic understanding on priming utterance processing. The paper closes with conclusions.

Background

The combinatorial nature of language provides us with virtually unlimited ways in which we can communicate meaning. This, of course, raises the question of how precisely an utterance should then be understood as it is being heard. Empirical studies in various branches of psycholinguistics and cognitive neuroscience have investigated what information listeners use when comprehending spoken utterances. An important observation across these studies is that interpretation *in context* plays a crucial role in the comprehension of utterance as it unfolds. Following (Knoeferle and Crocker, 2006) we can identify two core dimensions of the interaction between linguistic context and situated context. One is the *temporal dimension*. Attentional processes in situational perception appear to be closely time-locked with utterance comprehension. This can be witnessed by for example eye movements. The second is the *information dimension*. This indicates that listeners not only use linguistic information during utterance comprehension, but also scene understanding and "world knowledge." Below we discuss aspects of these dimensions in more detail.

Multi-level integration in language processing

Until the early 1990s, the dominant model of language comprehension was that of a modular, stage-like process; see for example (Fodor, 1983). On this model, a language user would sequentially construct each level of linguistic comprehension – from auditory recognition all the way to pragmatic, discourse-level interpretation. As (Van Berkum et al., 2003) observe, two hypotheses followed from this view. Firstly, people first construct a local, context-*independent* representation of the communicated meaning, before this meaning is interpreted against the preceding discourse context. Secondly, and related, is the hypothesis that discourse context-related processing only enters the process of language comprehension at a relatively late stage.

Opposing these hypotheses is the view that language comprehension is an incremental process, in which each level of linguistic analysis is performed in parallel. Every new word is immediately related to representations of the preceding input, across several levels – with the possibility for using the interpretation of a word at one level to co-constrain its interpretation at other levels. A natural prediction that follows from this view is that interpretation against dialogue context can in principle affect utterance comprehension *as the utterance is incrementally analyzed*, assisting in restricting the potential for grammatical forms of ambiguity. (Crain and Steedman, 1985; Altmann and Steedman, 1988) phrased this as a *principle of parsimony*: those grammatical analyses are selected that for their reference resolution impose the least presuppositional requirements on a dialogue context.

Since then, various studies have investigated further possible effects of dialogue context during utterance comprehension. Methodologically, psycholinguistic studies have primarily investigated the effects of dialogue context by measuring *saccadic eye movements* in a visual scene, based on the hypothesis that eye movements can be used as indications of underlying cognitive processes (Tanenhaus et al., 1994; Liversedge and Findlay, 2000). Alternatively, cognitive neuroscience-based studies use event-related brain potentials (ERPs) to measure the nature and time course of the effects of discourse context on human sentence comprehension (Van Berkum, 2004).

Both lines of study have found that lexical, semantic and discourse-level integrative effects occur in a closely time-locked fashion, starting already at the phoneme or sub-word level; (Allopenna et al., 1998), and (van Berkum et al., 1999b; Van Berkum et al., 2003; Van Petten et al., 1999). Particularly, a range of discourse-level integrative effects have observed. Referential binding has been shown to play a role in the constraining various types of local syntactic ambiguities, like garden path-constructions (Crain and Steedman, 1985; Altmann and Steedman, 1988; Altmann, 1988), and relative clauses (Spivey et al., 1993; Spivey and Tanenhaus, 1998); (van Berkum et al., 1999a; van Berkum et al., 1999b; Van Berkum et al., 2003). These effects primarily concern a *disambiguation* of already built structures. Integrating semantic and discourse-level information during utterance comprehension also has important *anticipatory* effects. (Tanenhaus et al., 2000; Dahan and Tanenhaus, 2004); (Van Berkum et al., 2005) observe how contextual information influences what lexical meanings can be anticipated, priming phonological understanding and lexical access. (Contextual information can even override disprefered lexical meaning (Nieuwland and Van Berkum, 2006).)

Anticipatory effects indicate that utterance comprehension is thus not only an incremental process of constructing and then disambiguating. Anticipation enables context-dependent phonological recognition, lexical re-

Proc. LangRo'2007
978-972-96895-2-9

trieval, and syntactic construction - without there being a need to generate and test all combinatory possible constructions. Incrementality and anticipation based on multi-level integration appears to give rise to a process in which comprehension arises through a convergence based on constraining and co-activation. Discourse context and the interpretative contexts which are delineated during utterance comprehension converge to become functionally identical (Van Berkum et al., 2003). As a result, ambiguity need not even arise, or is at least being much more limited a priori through context.

An important issue in all of the above remains of course the degree to which integrative effects indeed should commit to a certain understanding. Garden path sentences are a good example. They show that overcommitment risks the need for re-interpretation – an issue for *cognitive control* (Botvinick et al., 2001; Hommel et al., 2002; Novick et al., 2005).

Language processing and situational experience

We already noted before that human language processing integrates *linguistic* and *non-linguistic* information. Below we discuss studies which investigate how categorical and contextual information from situated experience can effect utterance comprehension. These studies use eye-trackers to monitor where people look at in a scene, and when.

(Altmann and Kamide, 1999) present a study revealing that listeners focus their attention on objects before these objects are referred to in the utterance. Figure 1 illustrates the setup of the study. When someone hears "The cat chases the mouse", her gaze already moves to the mouse in the scene before she has actually heard that word; similarly for "The mouse eats the cheese." Knowing that cats typically chase mice (not cheese), and that the argument structure of *chase* reflects this, the listener *expects* that the next object to be mentioned will be the mouse, and directs gaze to that object. We thus see an anticipatory effect arising from the online integration of lexico-semantic information (verbal argument structure), situational context (the present objects, and the intended action), and categorical knowledge (prototypical object-action relations).

Figure 1: Mouse, cheese, cat

Not only world knowledge can influence online utterance comprehension, also scene understanding can. For example, consider the situation in Figure 2. (Tanenhaus et al., 1994) show that, once the listener has heard "Put the apple on the towel ..." she faces the ambiguity of whether to put the (lone) apple onto the (empty) towel, or to take the apple that is on the towel and put it somewhere else. The ambiguity is revealed as visual search in the scene. Only once she has heard the continuation "... into the box" this ambiguity can be resolved. Interestingly, in (Tanenhaus et al., 1994) the listener cannot directly manipulate the objects. If this is possible (cf. Figure 2), (Chambers et al., 2004) show that also reachability plays a role in comprehending the utterance. Because only one apple is reachable, this is taken as the preferred referent, and as such receives the attention. This underlines the effect *physical embodiment* may have on language comprehension.

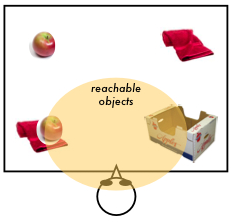


Figure 2: Put, apple, towel, box

Scene understanding also concerns the *temporal projection* towards possible future events (Endsley, 2000). (Altmann and Kamide, 2004; Kamide et al., 2003) show how such projection can also affect utterance comprehension. These studies used a scene with a table, and besides it a glass and a bottle of wine, as illustrated in Figure 3 (left). Investigated was where listeners look when they hear "The woman will put the glass on the table. Then, she will pick up the wine, and pour it carefully into the glass." It turns out that after hearing the "pouring" phrase, listeners look at the table, not the glass. Listeners thus explicitly project the result of the picking action into the scene, imagining the scene Figure 3 (right).

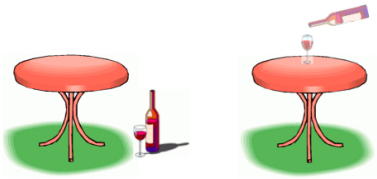


Figure 3: Pouring, wine, glass, table

These studies reveal that the interaction between vision and language is not *direct*, but *mediated* (Altmann and Kamide, 2004). Categorical understanding plays an important role in the sensorimotoric grounding of language. This is further underlined by studies like (Glenberg and Kaschak, 2002; De Vega et al., 2004), following up on the idea of category systems as mediating between perceptual modalities and language (Glenberg, 1997; Barsalou, 1999). These studies show how categorical understanding gives rise to expectations based on affordances, influencing comprehension of spatial or temporal aspects of action verbs.

In conversational dialogue (Hadelich and Crocker, 2006; Pickering and Garrod, 2004) gaze has been shown to be automatically aligned in simple collaborative interaction. The time intervals between eye-fixations during production and comprehension of a referring expression are shorter than in monologue. This is further evidence for the relevance of visual common ground of interlocutors and how that accelerates the activation of jointly relevant concepts.

Situated language processing in AI/HRI

Studies on how humans process visually situated dialogue show an important aspect of ”grounding” is based on how we can resolve a visual referent for an object reference. In establishing referents, listeners use visual and spatio-temporal properties of objects, and combine these properties with various forms of salience.

Several approaches have been proposed for visual referent resolution in human-robot interaction, in relation to language processing. Gorniak & Roy (Gorniak and Roy, 2004; Gorniak and Roy, 2005) present an approach in which utterance meaning is probabilistically mapped to visual and spatial aspects of objects in the current scene. Recently, they have extended their approach to include action-affordances (Gorniak and Roy, 2007). Their focus has primarily been on the grounding aspect, though. Although they use an incremental approach to constructing utterance meaning, grounding meanings in the social and physical context as they are construed, the (im)possibility to ground alternative meanings does not feed back into the incremental process to prune inviable analyses. This is where they differ from e.g. Scheutz et al (Scheutz et al., 2004; Brick and Scheutz, 2007). Scheutz et al present a model for incremental utterance processing in which the analyses are pruned if it is impossible to find visual referents for them.

Our approach to incremental language analysis is closely related to that of Scheutz et al. We incrementally build up a representation of utterance meanings, in parallel to syntactic analyses (Steedman, 2000). In this we (jointly) differ from other approaches such as (Allen et al., 1996; Mori et al., 2001; Rosé et al., 2002), who only build syntactic analyses. We advance on Scheutz et al in several ways, though. We analyze utterance meaning incrementally not only at the level of grammar, but also relative to the structure of the dialogue context. This allows different levels of linguistic description to constrain possible interpretations (Stone and Doran, 1997). Furthermore, we do not deal with individual analyses, but with a ”packed” representation (Oepen and Carroll, 2000; Carroll and Oepen, 2005) to handle linguistic ambiguity. Ambiguity is inherent in natural language. Often, parts of an utterance may be understood in different ways. Packing provides an efficient way to represent ambiguity. Parts shared across different analyses are represented only once, and ambiguities are reflected by different ways in which such parts can be connected. These packed representations are subsequently related to information about the (possibly dynamic) situation (Kruijff et al., 2006) and ongoing tasks (Allen et al., 2001; DeVault and Stone, 2003; Brenner et al., 2007; Gorniak and Roy, 2007). Should a possible meaning turn out to present an unresolvable reference, we discard it from the set of analyses the parser maintains.

Approach

Our approach has been implemented as part of an artificial cognitive architecture, built using the CoSy Architecture Schema Toolkit (CAST) (Hawes et al., 2007a; Hawes et al., 2007b). For the purpose of this paper, we focus on an architecture consisting of subsystems for visual and spatial processing of the situation, for interconnecting (”grounding”) content across subsystems, and for dialogue processing.

In CAST, we conceive of a cognitive architecture as a distributed collection of subsystems for information processing (Hawes et al., 2007a; Hawes et al., 2007b). Each subsystem consists of one or more processes, and a working memory. The processes can access sensors, effectors, and the working memory to share information within the subsystem. We divide processes into unmanaged, data-driven and managed, goal-driven processes. A data-driven process writes information onto the working memory in an ”unmanaged” fashion, typically whenever that information becomes available (e.g. from a sensor). A goal-driven process performs a specific type of interpretation of information available in working memory. This is a ”managed” process controlled by the subarchitecture’s task manager. The task manager decides when a goal process may, or may not, carry out its processing. This enables the subarchitecture to synchronize various forms of information processing.

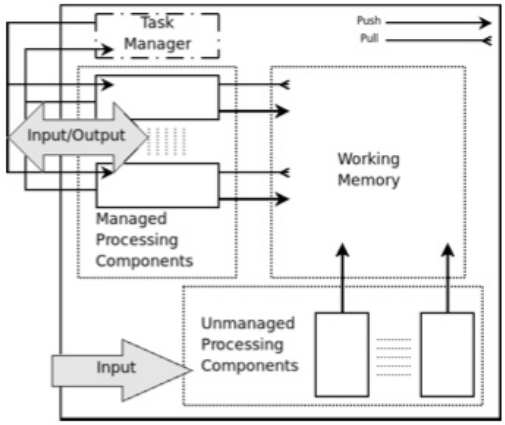


Figure 4: Abstract organization of a subarchitecture

Subsystems can also share information with other subsystems. We do this by monitoring a working memory of another subsystem, and reading/writing content to it.

Typically, a subsystem uses its own representation formats to deal most efficiently with the data it needs to handle. For example, the visual working memory contains regions of interest generated by a segmentor and proto-objects generated by interpreting these regions, whereas the dialogue subsystem contains logical forms generated from parsing utterances, and spatial reasoning maintains abstractions of physical objects with qualitative spatial relationships between them.

In our overall system, we have subsystems for vision, dialogue processing, manipulation, spatial reasoning (local scenes as well as multi-level maps), planning, coordination, and binding (used for symbol grounding). Several instantiations of this system have been described else-

Proc. LangRo'2007
978-972-96895-2-9

where (Hawes et al., 2007a; Brenner et al., 2007; Kruijff et al., 2007). Together, these subsystems create a system that can learn and communicate about objects and spatial locations with a user, and perform manipulation and navigation tasks.

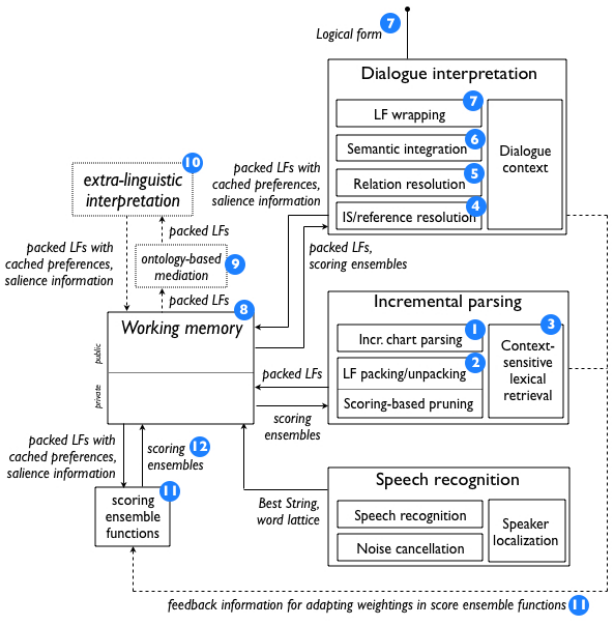


Figure 5: Dialogue processing (comprehension part)

Figure 5 illustrates the comprehension side of our dialogue processing subsystem.[1] (The numbers in the text refer to the round, blue labels in the figure.)

For speech recognition we use Nuance v8.5, to which the subsystem connects over a SIP connection. This enables us to use any number of microphones to "speak" to the robot – enabling both face-to-face and remote dialogue. Using an 8-microphone array on the robot we can do basic forms of noise cancellation and speaker localization. Speech recognition stores a recognition result on working memory in the form of a best string. Once this information becomes available, an incremental parsing process is triggered.

We have factorized (incremental) parsing into several, interconnected functions: the incremental parsing process itself (1), packing/unpacking and pruning of incrementally construed analyses of utterance meaning (2), and context-sensitive lexical retrieval (3). Parsing is based on a bottom-up Early chart parser (Sikkel, 1999) built for incrementally parsing Combinatory Categorial Grammar (Steedman, 2000; Baldridge and Kruijff, 2003). Its implementation relies on basic functionality provided by OpenCCG[2].

Incremental chart parsing creates partial, and integrated analyses for a string in a left-to-right fashion. As each word in the utterance is being scanned, the parser retrieves from the lexicon (3) a set of lexical entries. A lexical entry specifies for a word all its possible syntactic and semantic uses. During parsing, this information is used to integrate the word into possible analyses. By factorizing out lexical retrieval we have made it possible to use information about the situated- and task-context to restrict what lexical meanings are retrieved ("activated") for a word. After each word, the parser's chart maintains one or more possible analyses in parallel. These analyses represent the syntactic and semantic structure built for the utterance so far, and indicate possible ways in which these analyses can be continued by means of open arguments.

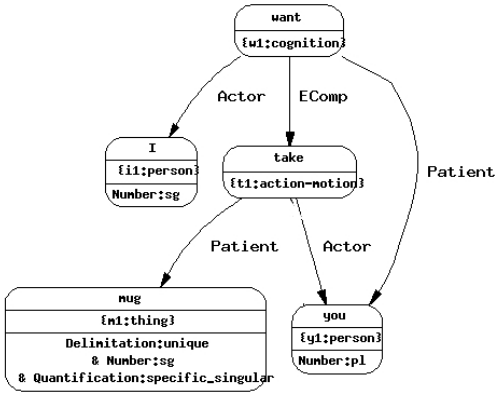


Figure 6: Logical form "I want you to take the mug"

Semantic structure is represented as an ontologically richly sorted, relational structure – a logical form (Baldridge and Kruijff, 2002). Figure 6 gives an example of a logical form (system output). Each node has a unique identifier with an associated ontological sort (e.g. $t1$ of sort $action-motion$), and a proposition (e.g. **take**). Nodes are connected through named relations. These indicate how the content of a single node contributes to the meaning of the whole expression. For example, "you" ($y1$) both indicates the one whom something is wanted of (*Patient*-relation from $w1$), and the one who is to perform the taking action (*Actor*-relation from $t1$). Nodes carry additional features, e.g. $i1$ identifies a singular person.

After each step in incremental parsing, the current set of logical forms is packed to create a more efficient representation for computing with logical forms (Oepen and Carroll, 2000; Carroll and Oepen, 2005). Figure 7 illustrates the development of the packed packed representation for "take the mug". At the first step ("take"), 6 logical forms are packed together, showing we essentially have two alternative interpretations: "take" as an action, and as part of the expression "take a look." The second step ("take the") makes it clear we only need to look at the action-interpretation. The possible meanings for the determiner is expressed at the node for the Patient. At this point we have an *overspecified* meaning. Although the delimination is unique, we cannot tell at this point whether we are dealing with a singular object, or a non-singular (i.e. plu-

[1] Most of the indicated processes have been implemented at the time of writing. Under construction are still *semantic integration* and *IS* i.e. information structure resolution.

[2] http://openccg.sf.net

ral) object – all we know it has to be one or the other. This becomes determined in the third step (”take the mug”).

Once the parser has created a packed representation, this is provided to the working memory. At this point, several processes for dialogue interpretation further interpret the representation, by providing discourse referents for the objects and events in the logical forms (4) and trying to connect the utterance to the preceding dialogue context in terms of rhetorical relations and dialogue moves (Asher and Lascarides, 2003). The resulting interpretations are related to the packed logical forms through ”caches”. A cache is a representation in which content is associated with other content, maintaining a mapping between unique keys in the two content representations. By using caches on top of the packed logical forms, we achieve a scalable approach for multi-level dialogue interpretation.

The packed logical forms, together with any dialogue-level interpretation of the content, is then provided to subsystems for extra-linguistic interpretation (8–10) (see below). The result of such interpretation is one or more preference orders over the interpretations representation by the packed logical forms. Technically, a scoring function is a partial order over substructures in packed logical forms. We can define ensembles over these functions to integrate their preferences, as e.g. suggested in (Kelleher, 2005) for salience functions. Before each next parsing step, packed logical forms are then pruned based on scoring ensembles, and the parse chart is updated.

In the architecture discussed here we rely for visual referent resolution on a grounding process called *binding*. The basic idea is illustrated in Figure 8. Each subsystem can have a binding monitor, which is a process that monitors the subsystem’s working memory. Every time the working memory contains content that could be connected to content in other modalities, the binding monitor translates this content using a mapping between the subsystem’s own representational formalism, and an *amodal* format used in the binding subsystem. This is based on the idea of ontology-mediated information fusion, cf. (Kruijff et al., 2006).

The resulting representation is then written to the working memory in the binding subsystem. There it acts as a *proxy* – namely, as a proxy for content in the originating subsystem. The binding subsystem now applies strategies to combine proxies with similar content, but coming from different subsystems. Proxies can then be combined form unions. The power of the binding mechanism is that we can use a mixture of early- and late-fusion, and represent content at any level of abstraction.

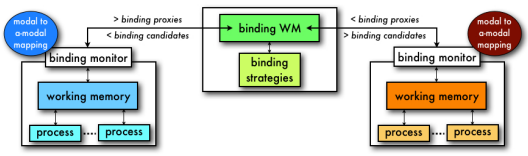


Figure 8: Cross-modal interconnectivity

Particularly, proxies from an individual subsystem can form relational structures. We thus can represent ”the blue mug” as a single proxy, as well as ”the blue mug next to the red box” as a relational structure connecting two proxies. Like individual proxies, the binder will try to connect relational structures – and either succeeding in doing so, e.g. if there is a blue mug next to the red box, or failing. This is crucial for situated dialogue processing (cf. also (Scheutz et al., 2004; Brick and Scheutz, 2007)).

Once we have a packed representation of logical forms, alternative relational structures are presented as proxies to the binding subsystem. By monitoring which relational structures can be bound into unions, and which ones cannot, we can prune the set of logical forms we maintain for the next step(s) in incremental parsing. We thus handle examples such as those discussed in (Brick and Scheutz, 2007) through an interaction between our binding subsystem, and the subsystem for dialogue processing.

Evaluation

Below we present preliminary results of an evaluation of the system. At the time of writing, we can only present statistical results evaluating the linguistic aspects of our processing model – not for the impact cross-modal binding has on linguistic processing. We do describe below how we will be able to statistically evaluate the impact of binding.

Design & measures

We have designed a set of eleven visual scenes, in which we can systematically vary the potential ambiguity of a visual object relative to specific types of referring expressions. Figure 9 gives an example of such a scene. Assuming we are looking at the scene from the robot’s viewpoint, expressions such as ”the blue thing” or ”the blue ball” uniquely refer to the blue ball (with identifier *b*2). If we furthermore take e.g. visual and topokinetic salience into account, the referring expression ”the mug” in ”take the mug” has a strong preference for the red mug (*m*1) as being the visual referent (the white mug (*m*2) being less visually salient, and unreachable).

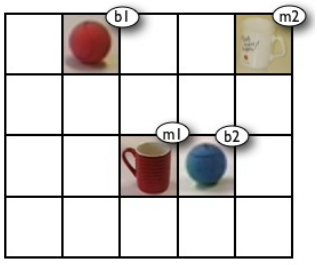


Figure 9: Sample visual scene

For these scenes, we have formulated a total of 58 utterances. These utterances express either commands (”put the mug to the left of the ball”) or assertions (”the mug is red”). The utterances vary in length, with a distribution

Proc. LangRo'2007
978-972-96895-2-9

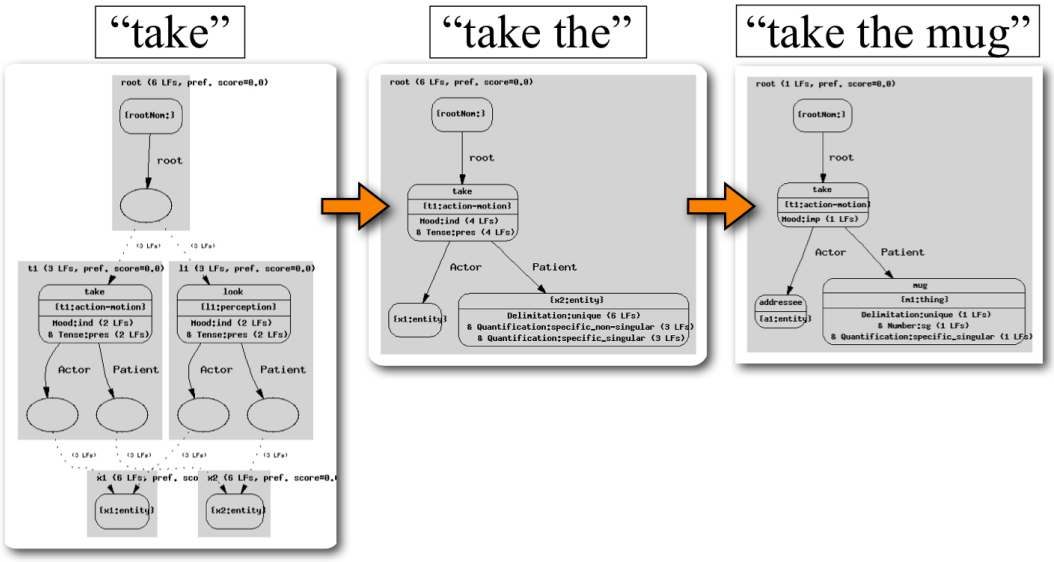


Figure 7: Example packed logical forms

as given in Table 1. The (weighted) average length of the utterances in the evaluation is 6.07 words.

The utterances include referring expressions, which may be ambiguous relative to the scene for which they have been formulated. This enables us to investigate the interplay between different forms of ambiguity. First, we want to explore to what degree we can resolve purely linguistic ambiguity (notably, syntactic PP-attachment ambiguities) against non-ambiguous situations. Second, we want to evaluate to what degree ambiguity in situation awareness can be resolved through non-ambiguous linguistic meaning – or, if both would be ambiguous, to what degree we can still reduce the ambiguity. By systematically varying the ambiguity in the scenes, and in the structure of the utterances, we can properly evaluate these factors.

length	16	14	13	12	11
# utterances	1	2	2	4	7

length	10	9	8	7	6	5	4
# utterances	4	3	5	4	5	11	10

Table 1: Distribution of #utterances over lengths

In the experiment, we have used two incremental parsers. One is the incremental parser which uses grammatical knowledge to prune analyses during parsing (”pruning”). The other parser does not do any pruning, and functions as baseline (”baseline”). Below we show results of the pruning parser relative to the baseline performance.

Results

We present here results that measure the improvements the pruning parser makes over the baseline in terms of number of final analyses, the size of the resulting packed logical form, and time to obtain all complete analyses. The first two aspects measure memory use. Memory use is a factor that has an important impact on situated grounding of language. The fewer analyses, and the smaller the packed logical form, the less *varying* (or ambiguous) information we need to try and bind to information in other modalities.

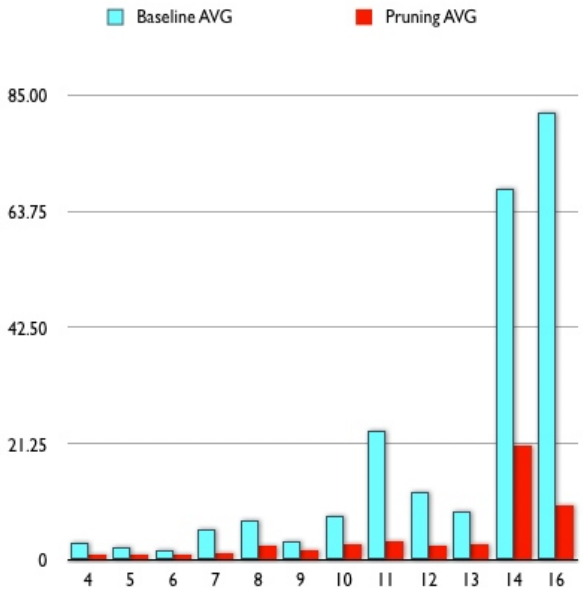


Figure 10: Sentence length (X) * Number of final analyses (baseline, pruning) (Y)

Figure 10 shows a bar chart of the number of final analyses produced by the baseline parser (light-blue, left) and the pruning parser (red, right). Using weighted averages, we get a 65.92% improvement of the pruning parser over the baseline. This improvement is statistically significant (one-way analysis of variance, F value = 27.036, Pr $>$ 0.001).

If we look at the variation in size of the packed logi-

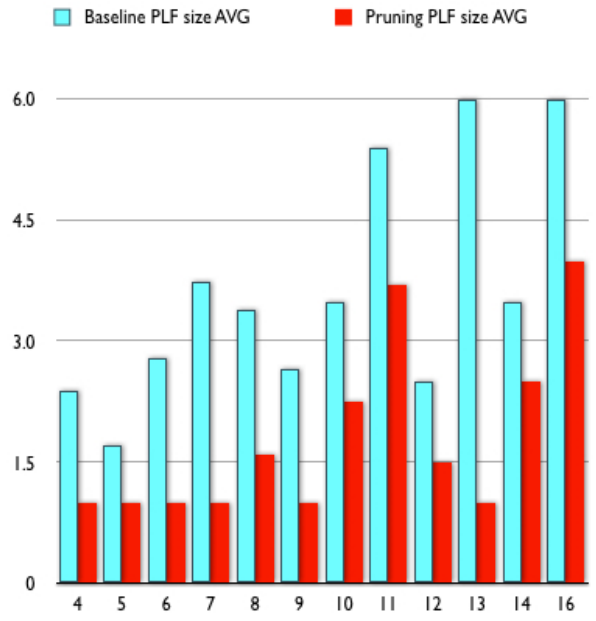


Figure 11: Sentence length (X) * Number of final packed LF size (baseline, pruning)(Y)

cal forms, we see a similar improvement. Figure 11 plots the sizes of the resulting packed logical forms against the utterance length, for the two parsers. This shows a 49.87% improvement of the pruning parser over the baseline (weighted average). Again, this result is statistically significant (one-way analysis of variance, F value=6.5283, Pr $>$0.01).

Figure 12 gives the results for time to parse completion, for the pruning parser and the baseline. On a weighted average, the pruning parser presents a 6.04% over the baseline (statistically significant, F value = 115.40, Pr $>$ 0.001).

Discussion

The results show improvements of the pruning parser over the baseline in terms of memory use, and in time to completion. We have obtained these improvements on a data set of 58 utterances of varying complexity – not on isolated examples – and shown them to be statistically significant.

These results are in and by themselves not surprising – if a parser does pruning, it should do better than a baseline which does not. What is more interesting in the light of situated dialogue processing is that, even when we do use grammatical knowledge to select analyses, this may still not be enough to reduce the final number of analyses to 1. If that were the case, then there would be no need to use grounding in the situation. On the data set we have used, we have a (weighted) average of 2.71 final analyses for the pruning parser (against a weighted average of 10.77 for the baseline).

Our next step is to evaluate our system, including the visual scenes on which the utterances have been formulated. The system enables us to prune analyses based on

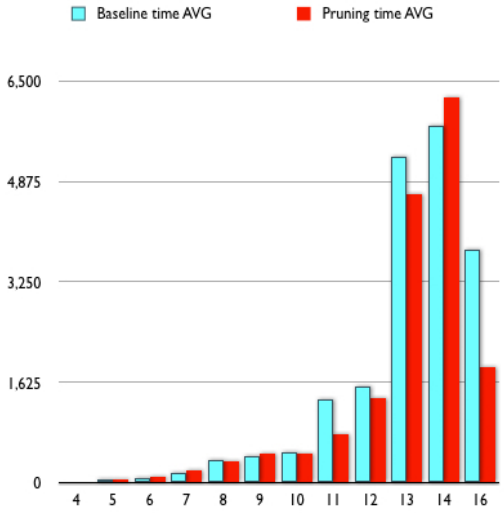


Figure 12: Sentence length (X) * Time (ms) to parse completion (baseline, pruning)(Y)

what content in a packed logical form can (or cannot) be grounded in situation awareness.

Conclusions

We presented work on an implemented model of situated dialogue processing. The model is based on the idea that to understand situated dialogue, linguistic meaning needs to be coupled to the situated context. Processing dialogue incrementally, information about the dialogue- and situated context can help at each step to focus the linguistic analysis. The implemented has been evaluated on a data set of 58 utterances formulated on 11 different visual scenes. Investigating the effects of using linguistic knowledge, the results show that using such knowledge can greatly improve the performance of an incremental parser, but cannot fully reduce linguistic ambiguity. This confirms the need for including information about the situated context to further reduce that ambiguity. We are currently planning follow-up evaluations that will investigate these effects further.

Acknowledgements

The research reported of in this paper was supported by the EU FP6 IST Cognitive Systems Integrated project *Cognitive Systems for Cognitive Assistants* "CoSy" FP6-004250-IP.

References

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001). Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–37.

Allen, J., Miller, B., Ringger, E., and Sikorski, T. (1996). A robust system for natural spoken dialogue. In *Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL'96)*, pages 62–70.

Allopenna, P., Magnuson, J., and Tanenhaus, M. (1998). Tracking the time course of spoken word recognition using eye

movements: Evidence for continuous mapping models. *Journal of Memory and Language*, 38(4):419–439.

Altmann, G. and Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3):247–264.

Altmann, G. and Kamide, Y. (2004). Now you see it, now you don't: Mediating the mapping between language and the visual world. In Henderson, J. and Ferreira, F., editors, *The Interface of Language, Vision, and Action: Eye Movements and The Visual World*, pages 347–386. Psychology Press, New York NY.

Altmann, G. M. (1988). Ambiguity in sentence processing. *Trends in Cognitive Sciences*, 2(4).

Altmann, G. T. and Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Asher, N. and Lascarides, A. (2003). *Logics of Conversation*. Cambridge University Press, Cambridge, UK.

Baldridge, J. and Kruijff, G. (2002). Coupling CCG and hybrid logic dependency semantics. In *Proc. ACL 2002*, pages 319–326, Philadelphia, PA.

Baldridge, J. and Kruijff, G.-J. M. (2003). Multi-modal combinatory categorial grammmar. In *Proceedings of EACL'03*, Budapest, Hungary.

Barsalou, L. (1999). Perceptual symbol systems. *Behavioral & Brain Sciences*, 22:577–660.

Botvinick, M., Braver, T., Barch, D., Carter, C., and Cohen, J. (2001). Conflict monitoring and cognitive control. *Psychological Review*, 108(3):624–652.

Brenner, M., Hawes, N., Kelleher, J., and Wyatt, J. (2007). Mediating between qualitative and quantitative representations for task-orientated human-robot interaction. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*.

Brick, T. and Scheutz, M. (2007). Incremental natural language processing for HRI. In *Proceeding of the ACM/IEEE international conference on Human-Robot Interaction (HRI'07)*, pages 263 – 270.

Carroll, J. and Oepen, S. (2005). High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP'05)*, pages 165–176.

Chambers, C., Tanenhaus, M., and Magnuson, J. (2004). Actions and affordances in syntactic ambiguity resolution. *Jnl. Experimental Psychology*, 30(3):687–696.

Crain, S. and Steedman, M. (1985). On not being led up the garden path: The use of context by the psychological syntax processor. In Dowty, D. R., Karttunen, L., and Zwicky, A. M., editors, *Natural language parsing: Psychological, computational, and theoretical perspectives*. Cambridge University Press.

Dahan, D. and Tanenhaus, M. (2004). Continuous mapping from sound to meaning in spoke-language comprehension: Immediate effects of verb-based thematic constraints. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(2):498–513.

De Vega, M., Robertson, D., Glenberg, A., Kaschak, M., and Rinck, M. (2004). On doing two things at once: Temporal constraints on actions in language comprehension. *Memory and Cognition*, 32(7):1033–1043.

DeVault, D. and Stone, M. (2003). Domain inference in incremental interpretation. In *Procedings of the Fourth Workshop on Inference in Computational Semantics (ICOS'04)*.

Endsley, M. (2000). Theoretical underpinnings of situation awareness: A critical review. In Endsley, M. R. and Garland, D. J., editors, *Situation awareness analysis and measurement*. Lawrence Erlbaum.

Fodor, J. (1983). *The Modularity of Mind*. The MIT Press, Cambridge MA.

Glenberg, A. (1997). What memory is for. *Behavioral & Brain Sciences*, 20:1–55.

Glenberg, A. and Kaschak, M. (2002). Grounding language in action. *Psychonomic Bulletin & Review*, 9(3):558–565.

Gorniak, P. and Roy, D. (2004). Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470.

Gorniak, P. and Roy, D. (2005). Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005)*.

Gorniak, P. and Roy, D. (2007). Situated language understanding as filtering perceived affordances. *Cognitive Science*, 31(2):197–231.

Hadelich, K. and Crocker, M. (2006). Gaze alignment of interlocutors in conversational dialogues. In *Proc. 19th CUNY Conference on Human Sentence Processing*, New York, USA.

Hawes, N., Sloman, A., Wyatt, J., Zillich, M., Jacobsson, H., Kruijff, G., Brenner, M., Berginc, G., and Skocaj, D. (2007a). Towards an integrated robot with multiple cognitive functions. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI'07)*.

Hawes, N., Zillich, M., and Wyatt, J. (2007b). BALT & CAST: Middleware for cognitive robotics. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, pages 998 – 1003.

Hommel, B., Ridderinkhof, K., and Theeuwes, J. (2002). Cognitive control of attention and action: Issues and trends. *Psychological Research*, 66:215–219.

Kamide, Y., Altmann, G., and Haywood, S. (2003). The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye-movements. *Jnl. Memory and Language*, 49(1):133–156.

Kelleher, J. (2005). Integrating visual and linguistic salience for reference resolution. In *Proceedings of the 16th Irish conference on Artificial Intelligence and Cognitive Science (AICS-05)*.

Knoeferle, P. and Crocker, M. (2006). The coordinated interplay of scene, utterance, and world knowledge: evidence from eye tracking. *Cognitive Science*.

Kruijff, G., Kelleher, J., and Hawes, N. (2006). Information fusion for visual reference resolution in dynamic situated dialogue. In André, E., Dybkjaer, L., Minker, W., Neumann, H., and Weber, M., editors, *Perception and Interactive Technologies (PIT 2006)*. Spring Verlag.

Kruijff, G., Zender, H., Jensfelt, P., and Christensen, H. (2007). Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems, Special Issue on Human and Robot Interactive Communication*, 4(1):125–138.

Liversedge, S. and Findlay, J. (2000). Saccadic eye movements and cognition. *Trends in Cognitive Science*, 4(1):6–14.

Mori, D., Matsubara, S., and Inagaki, Y. (2001). Incremental parsing for interactive natural language interface. In *2001 IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 2880–2885.

Nieuwland, M. and Van Berkum, J. (2006). When peanuts fall in love: N400 evidence for the power of discourse. *Journal of Cognitive Neuroscience*, 18(7):1098–1111.

Novick, J., Trueswell, J., and Thompson-Schill, S. (2005). Cognitive control and parsing: Reexamining the role of Broca's area in sentence comprehension. *Cognitive, Affective, and Behavioral Neuroscience*, 5(3):263–281.

Oepen, S. and Carroll, J. (2000). Ambiguity packing in constraint-based parsing: Practical results. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000)*, pages 162–169.

Pickering, M. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–225.

Rosé, C. P., Roque, A., Bhembe, D., and VanLehn, K. (2002). An efficient incremental architecture for robust interpretation. In *Proceedings of the Human Languages Technologies Conference*.

Scheutz, M., Eberhard, K., and Andronache, V. (2004). A real-time robotic model of human reference resolution using visual constraints. *Connection Science Journal*, 16(3):145–167.

Sikkel, K. (1999). *Parsing Schemata*. Springer Verlag.

Spivey, M. and Tanenhaus, M. (1998). Syntactic ambiguity resolution in discourse: Modeling the effects of referential context and lexical frequency. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24:15211543.

Spivey, M., Trueswell, J., and Tanenhaus, M. (1993). Context effects in syntactic ambiguity resolution: discourse and semantic influences in parsing reduced relative clauses. *Canadian Journal of Experimental Psychology*, 47(2):276–309.

Steedman, M. (2000). *The Syntactic Process*. The MIT Press, Cambridge MA.

Stone, M. and Doran, C. (1997). Sentence planning as description using tree-adjoining grammar. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, pages 198–205.

Tanenhaus, M., Magnuson, J., Dahan, D., and Chambers, G. (2000). Eye movements and lexical access in spoken-language comprehension: Evaluating a linking hypothesis between fixations and linguistic processing. *Journal of Psycholinguistic Research*, 29(6):557–580.

Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., and Sedivy, J. (1994). Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.

Van Berkum, J. (2004). Sentence comprehension in a wider discourse: Can we use erps to keep track of things? In Carreiras, M. and Jr., C. C., editors, *The on-line study of sentence comprehension: Eyetracking, ERPs and beyond*, pages 229–270. Psychology Press, New York NY.

van Berkum, J., Brown, C., and Hagoort, P. (1999a). Early referential context effects in sentence processing: Evidence from event-related brain potentials. *Journal of Memory and Language*, 41:147–182.

Van Berkum, J., Brown, C., Zwitserlood, P., Kooijman, V., and Hagoort, P. (2005). Anticipating upcoming words in discourse: Evidence from erps and reading times. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 31(3):443–467.

van Berkum, J., Hagoort, P., and Brown, C. (1999b). Semantic integration in sentences and discourse: Evidence from the n400. *Journal of Cognitive Neuroscience*, 11(6):657–671.

Van Berkum, J., Zwitserlood, P., Brown, C., and Hagoort, P. (2003). When and how do listeners relate a sentence to the wider discourse? evidence from the n400 effect. *Cognitive Brain Research*, 17:701–718.

Van Petten, C., Coulson, S., Rubin, S., Plante, E., and Parks, M. (1999). Time course of word identification and semantic integration in spoken language. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(2):394–417.

Producing Believable Robot Gaze When Comprehending Visually Situated Dialogue

Geert-Jan M. Kruijff[1] and Maria Staudte[2]

[1]Language Technology Lab, German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken Germany

[2]Dept. of Computational Linguistics, Saarland University, Saarbrücken Germany

gj@dfki.de

Abstract

The paper presents an implemented approach to producing robot gaze during comprehending visually situated dialogue. The approach is based on an incremental model for processing situated dialogue. In this model, utterance interpretations are build step-by-step, in a "left-to-right" fashion. At each step, grammatical and dialogue-level information is combined with information about the visually situated context. As a consequence, utterance processing can be guided so as to construct only situationally appropriate interpretations. Furthermore, at each step a set of visual referents is determined, to which the unfolding utterance meaning is currently making reference. In the approach, this information is used to drive robot gaze, letting the robot change its fixation onto the most recent visual referent. The underlying assumption is that gaze behavior helps to establish joint attention ("common ground") in a dialogue, if there is congruency between where the robot is looking, and what the (intended) visual referent is. The paper reports on a pilot study in which this assumption is studied. The results show statistically significant interactions between congruence, believability, and appropriateness of referring expression.

Introduction

In situated dialogue, people not only talk – they also look. They look at the visual objects they believe are being referred to. This serves a fundamental function in dialogue. By aligning what they attend to in the visual context, the resulting *joint attention* indicates that they share the same understanding of what is being talked about (Garrod and Pickering, 2004; Pickering and Garrod, 2004).

In this paper, we discuss an implemented approach that makes a robot produce similar behavior when it is trying to understand an utterance. Several empirical studies have confirmed that such robot behavior would make human-robot interaction more natural (Breazeal et al., 2004a; Miyauchi et al., 2004; Sidner et al., 2004; Sidner et al., 2005; Yoshikawa et al., 2006). It is still an open question though how to produce such behavior in a way that it really takes the situation into account. Current approaches primarily rely on scripted behaviors which are not grounded in the visual context.

The approach we present relies on explicitly grounding dialogue in the situated context. The main idea is to use an *incremental* model for dialogue analysis, and step-by-step connect the linguistic representations with information about the visually situated context. From this interconnection we can then derive what the visual objects are that are being talked about, and so drive the robot's *gaze* – i.e. what objects it should fixate on, and when it should move from looking at one object to the next. We use insights from psycholinguistics in postulating what factors in the visually situated context *might* play a role (Altmann and Steedman, 1988; Altmann and Kamide, 2004; Knoeferle and Crocker, 2006). We have performed a pilot study to empirically evaluate our approach.

Our approach is related to other recent work on incremental language processing for dialogue systems (Allen et al., 1996; Mori et al., 2001; Rosé et al., 2002), and for human-robot interaction (Brick and Scheutz, 2007). Like (Brick and Scheutz, 2007) we analyze an utterance for its meaning, not just for syntactic structure (Allen et al., 1996; Mori et al., 2001; Rosé et al., 2002). We advance on (Brick and Scheutz, 2007) by analyzing utterance meaning incrementally also relative to the structure of the dialogue context, allowing different levels of linguistic description to constrain possible interpretations (Stone and Doran, 1997). We adopt a "packed" representation of the linguistic analyses (Oepen and Carroll, 2000; Carroll and Oepen, 2005) to efficiently handle alternative (i.e. ambiguous) meanings. These packed representations are subsequently related to information about the situation and ongoing tasks (Allen et al., 2001; DeVault and Stone, 2003; Gorniak and Roy, 2007). This essentially comes down to trying to resolve how a meaning refers to the current context (Stone and Doran, 1997; Brick and Scheutz, 2007) – intuitively, if a meaning presents an unresolvable reference, it can be discarded. Whenever a step in the incremental utterance analysis introduces a new object in the utterance meaning, we thus get a set of possible visual referents for that object description. The basic idea in producing "gaze" is to let the robot look (i.e. fixate) at the visual referent(s) for the most recently added object(s).

An overview of the paper is as follows. We first provide further background to our approach. We discuss relevant psycholinguistic insights in what factors tend to influence understanding situated language, and position our approach in more detail to the current state-of-the-art. We then present our approach in detail. We discuss the cognitive architecture schema we employ (Hawes et al., 2007a; Hawes et al., 2007b), our incremental approach to multi-

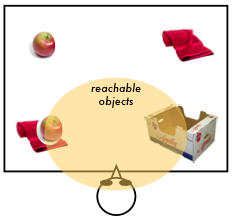
level dialogue analysis, how utterance meaning gets interconnected with the broader context – and how gaze behavior is driven by the resolution of the visual objects that we find the utterance refers to, as we incrementally analyse its possible meanings. Finally, we discuss the results of a pilot experiment we have performed. The pilot investigates the extent to which a user believes the robot has understood what was said on the basis of where the robot looks. Results show statistically significant interactions between believability, and congruency between gaze and intended visual referent. The paper ends with conclusions.

Background

Empirical studies in psycholinguistics have investigated what information listeners use when comprehending spoken utterances. These studies use eye-trackers to monitor where people look at in a scene, and when. Knoeferle & Crocker (Knoeferle and Crocker, 2006) argue that these findings identify two core dimensions of the interaction between language and situated experience. One is the *temporal dimension*: Eye movements during utterance comprehension reveal that visual attention is closely time-locked with utterance comprehension. The second is the *information dimension*, indicating how for utterance comprehension listeners draw not only upon linguistic information, but also upon scene understanding and "world knowledge." Below we discuss studies investigating the latter two aspects.

Altmann & Kamide (Altmann and Kamide, 1999) show that listeners use "world knowledge" to anticipate what will be mentioned next in an utterance. They focus their attention on objects before these objects are explicitly referred to. For example, when someone hears "The cat chases the mouse.", her gaze already moves to the mouse in the scene before she has actually heard that word. Knowing that cats typically chase mice (not cheese), and that the argument structure of *chase* reflects this, the listener *expects* that the next object to be mentioned will be the mouse, and fixates on that object.

Also scene understanding influences how we understand an utterance. For example, consider the figure to the right. Tanenhaus et al (Tanenhaus et al., 1994) show that once the listener has heard "Put the apple on the towel ..." she faces the ambiguity of whether to put the (lone) apple onto the (empty) towel, or to take the apple that is on the towel and put it somewhere else. The ambiguity is revealed as visual search in the scene. Only once she has heard the continuation "... into the box" this ambiguity can be resolved. In (Tanenhaus et al., 1994) the listener cannot directly manipulate the objects. If this is possible, Chambers et al (Chambers et al., 2004) show that also reachability plays a role. Because the listener can only grasp the apple that is on the towel, this is taken as the preferred referent.



These the studies thus show that gaze fixations are derived from how we can resolve a visual referent for an object reference. In establishing referents, listeners use visual and spatial properties of objects, combined with visual salience and "topokinetic" salience derived from object reachability.

Several approaches have been proposed for visual referent resolution in human-robot interaction, in relation to language processing. Gorniak & Roy (Gorniak and Roy, 2004; Gorniak and Roy, 2005) present an approach in which utterance meaning is probabilistically mapped to visual and spatial aspects of objects in the current scene. Recently, they have extended their approach to include action-affordances (Gorniak and Roy, 2007). Their focus has primarily been on the grounding aspect, though. Although they use an incremental approach to constructing utterance meaning, grounding meanings in the social and physical context as they are construed, the (im)possibility to ground alternative meanings does not feed back into the incremental process to prune inviable analyses. This is where they differ from e.g. Scheutz et al (Scheutz et al., 2004; Brick and Scheutz, 2007). They present a model for incremental utterance processing in which the analyses are pruned if it is impossible to find visual referents for them.

Our approach to incremental language analysis is closely related to that of Scheutz et al. We incrementally build up a representation of utterance meanings, in parallel to syntactic analyses (Steedman, 2000). In this we (jointly) differ from other approaches such as (Allen et al., 1996; Mori et al., 2001; Rosé et al., 2002), who only build syntactic analyses. We advance on Scheutz et al in several ways, though. We analyze utterance meaning incrementally not only at the level of grammar, but also relative to the structure of the dialogue context. This allows different levels of linguistic description to constrain possible interpretations (Stone and Doran, 1997). Furthermore, we do not deal with individual analyses, but with a "packed" representation (Oepen and Carroll, 2000; Carroll and Oepen, 2005) to handle linguistic ambiguity. Ambiguity is inherent in natural language – often, parts of an utterance may be understood in different ways. Packing provides an efficient way to represent ambiguity. Parts shared across different analyses are represented only once, and ambiguities are reflected by different ways in which such parts can be connected. These packed representations are subsequently related to information about the (possibly dynamic) situation (Kruijff et al., 2006) and ongoing tasks (Allen et al., 2001; DeVault and Stone, 2003; Brenner et al., 2007; Gorniak and Roy, 2007). Should a possible meaning turn out to present an unresolvable reference, we discard that analysis from the set of analyses maintained by the parser.

We use this approach to incremental language processing as the basis for producing gaze fixations and -movements. The basic idea is simple. Whenever a new object description is introduced in the unfolding utterance meaning, we determine the set of possible visual referents. We then let the robot fixate at the visual referent(s)

Proc. LangRo'2007
978-972-96895-2-9

for the most recently added object(s). Although simple, this approach sets us apart from several other approaches to producing gaze in human-robot interaction. Most approaches adopt fixed scripted behaviors to drive gaze (Sidner et al., 2004), or make the robot look at an ”area of change” to signal understanding (Breazeal et al., 2004b). Alternatively, the robot is made to exactly mimic its human partner (Yoshikawa et al., 2006). The problem with these systems is that gaze is not produced on the basis of a deeper understanding of the situation, and how dialogue refers to that situation. This results in a rigid and merely reactive behavior that is not flexible enough to adapt to novel situations.

The approach to producing robot gaze we propose here is a natural extension of an incremental model of situated dialogue processing. Incrementally construed linguistic meaning gradually becomes grounded in the social and physical context in which the dialogue takes place, in ways that reflect the unique and dynamic nature of situations.

Approach

We have implemented our approach in an cognitive architecture based on the CoSy Architecture Schema Toolkit (CAST) (Hawes et al., 2007a; Hawes et al., 2007b). For the purpose of this paper, we focus on an architecture consisting of subsystems for visual and spatial processing of the situation, for interconnecting (”grounding”) content across subsystems, and for gaze and dialogue processing.

Cognitive architecture

In CAST, we conceive of a cognitive architecture as a distributed collection of subsystems for information processing (Hawes et al., 2007a; Hawes et al., 2007b). Each subsystem consists of one or more processes, and a working memory. The processes can access sensors, effectors, and the working memory to share information within the subsystem. Subsystems can also share information with other subsystems. Principally, this can be done by monitoring a working memory of another subsystem, and reading/writing content to it.

Typically, a subsystem establishes its own representation formats to deal most efficiently with the data it needs to handle. For example, the visual working memory contains regions of interest generated by a segmentor and proto-objects generated by interpreting these regions, whereas the dialogue subsystem contains logical forms generated from parsing utterances, and spatial reasoning maintains abstractions of physical objects with qualitative spatial relationships between them.

In our overall system, we have subsystems for vision, dialogue processing, manipulation, spatial reasoning (local scenes as well as multi-level maps), planning, coordination, and binding (used for symbol grounding). Several instantiations of this system have been described elsewhere (Hawes et al., 2007a; Kruijff et al., 2007). Together, these subsystems create a system that can learn and communicate about objects and spatial locations with a user, and perform manipulation and navigation tasks.

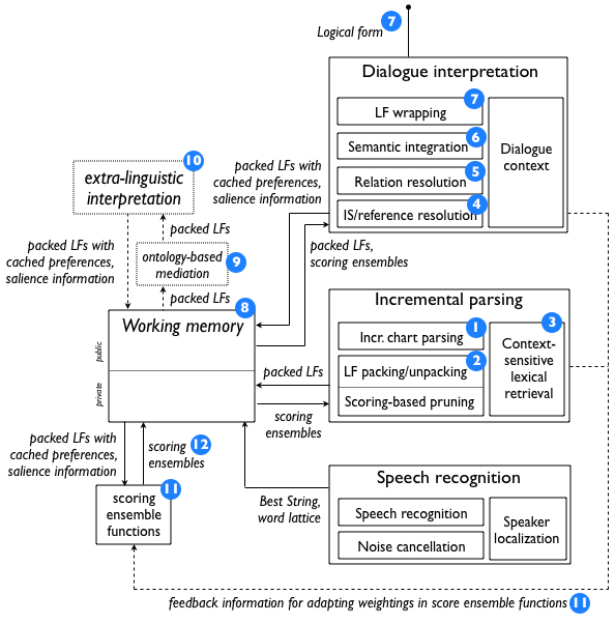


Figure 1: Dialogue processing (comprehension part)

Dialogue analysis

Figure 1 illustrates the comprehension side of our dialogue processing subsystem.[1] (The numbers in the text refer to the round, blue labels in the figure.)

For speech recognition we use Nuance v8.5, to which the subsystem is connects over a SIP connection. This enables us to use any number of microphones to ”speak” to the robot – enabling both face-to-face and remote dialogue. Using an 8-microphone array on the robot we can do basic forms of noise cancellation and speaker localization. Speech recognition stores a recognition result on working memory in the form of a best string. Once this information becomes available, an incremental parsing process is triggered.

We have factorized (incremental) parsing into several, interconnected functions: the incremental parsing process itself (1), packing/unpacking and pruning of incrementally construed analyses of utterance meaning (2), and context-sensitive lexical retrieval (3). Parsing is based on a bottom-up Early chart parser (Sikkel, 1999) built for incrementally parsing Combinatory Categorial Grammar (Steedman, 2000; Baldridge and Kruijff, 2003). Its implementation relies on basic functionality provided by OpenCCG[2].

Incremental chart parsing creates partial, and integrated analyses for a string in a left-to-right fashion. As each word in the utterance is being scanned, the parser retrieves from the lexicon (3) a set of lexical entries. A lexicon entry specifies for a word all its possible syntactic and semantic uses. During parsing, this information is used to integrate

[1] Most of the indicated processes have been implemented at the time of writing. Under construction are still *semantic integration* and *IS* i.e. information structure resolution.

[2] http://openccg.sf.net

the word into possible analyses. By factorizing out lexical retrieval we have made it possible to use information about the situated- and task-context to restrict what lexical meanings are retrieved (”activated”) for a word. After each word, the parser’s chart maintains one or more possible analyses in parallel. These analyses represent the syntactic and semantic structure built for the utterance so far, and indicate possible ways in which these analyses can be continued by means of open arguments.

Semantic structure is represented as an ontologically richly sorted, relational structure – a logical form (Baldridge and Kruijff, 2002). After each step in incremental parsing, the current set of logical forms is packed to create a more efficient representation for computing with logical forms (Oepen and Carroll, 2000; Carroll and Oepen, 2005). Figure 2 illustrates a packed representation of intermediate logical forms for ”put the ball to the left of the box”, packing together 30 logical forms.

Once the parser has created a packed representation, this is provided to the working memory. At this point, several processes for dialogue interpretation further interpret the representation, by providing discourse referents for the objects and events in the logical forms (4) and trying to connect the utterance to the preceding dialogue context in terms of rhetorical relations and dialogue moves (Asher and Lascarides, 2003). The resulting interpretations are related to the packed logical forms through ”caches”. A cache is a representation in which content is associated with other content, maintaining a mapping between unique keys in the two content representations. By using caches on top of the packed logical forms, we achieve a scalable approach for multi-level dialogue interpretation.

The packed logical forms, together with any dialogue-level interpretation of the content, is then provided to subsystems for extra-linguistic interpretation (8–10) (see §). The result of such interpretation is one or more preference orders over the interpretations representation by the packed logical forms. Technically, a scoring function is a partial order over substructures in packed logical forms. We can define ensembles over these functions to integrate their preferences, as e.g. suggested in (Kelleher, 2005) for salience functions. Before each next parsing step, packed logical forms are then pruned based on scoring ensembles, and the parse chart is updated.

Resolving referents

In the architecture discussed here we rely for visual referent resolution on a grounding process called *binding*. The basic idea is illustrated in Figure 3. Each subsystem can have a binding monitor, which is a process that monitors the subsystem’s working memory. Every time the working memory contains content that could be connected to content in other modalities, the binding monitor translates this content using a mapping between the subsystem’s own representational formalism, and an *amodal* format used in the binding subsystem. This is based on the idea of ontology-mediated information fusion, cf. (Kruijff et al., 2006).

The resulting representation is then written to the working memory in the binding subsystem. There it acts as a *proxy* – namely, as a proxy for content in the originating subsystem. The binding subsystem now applies strategies to combine proxies with similar content, but coming from different subsystems. Proxies that can be combined form unions. The power of the binding mechanism is that we can use a mixture of early- and late-fusion, and represent content at any level of abstraction.

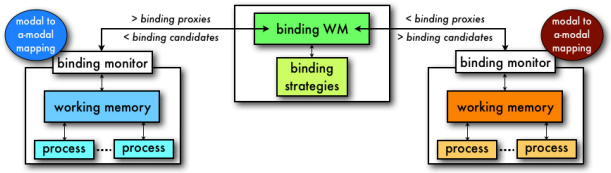


Figure 3: Cross-modal interconnectivity

Particularly, proxies from an individual subsystem can form relational structures. We thus can represent ”the blue mug” as a single proxy, as well as ”the blue mug next to the red box” as a relational structure connecting two proxies. Like individual proxies, the binder will try to connect relational structures – and either succeeding in doing so, e.g. if there is a blue mug next to the red box, or failing. This is crucial for situated dialogue processing (cf. also (Scheutz et al., 2004; Brick and Scheutz, 2007)).

Once we have a packed representation of logical forms, alternative relational structures are presented as proxies to the binding subsystem. By monitoring which relational structures can be bound into unions, and which ones cannot, we can prune the set of logical forms we maintain for the next step(s) in incremental parsing. We thus handle examples such as those discussed in (Brick and Scheutz, 2007) through an interaction between our binding subsystem, and the subsystem for dialogue processing.

Producing gaze

The result of binding is that we obtain, after each incremental interpretation step, a set of one or more visual referents for the objects represented by the packed logical forms. Depending on whether binding is able to resolve any syntactic ambiguities (as in e.g. ”put the apple on the towel ... ”), the set of referents may present referential ambiguity, or not.

The gaze subsystem monitors the binding working memory for unions of recently added proxies coming from the dialogue subsystem, bound to visual entities. Based on the (un)ambiguity of these unions, and the completeness of the linguistic analyses, the gaze subsystem will then produce one of the following behaviors:

Saccade from speaker to unambiguous visual referent, fixation: At the start of hearing a new utterance, the robot is looking at the user. As soon as the first visual referent is established, the robot moves to look at the visual referent,

Proc. LangRo'2007
978-972-96895-2-9

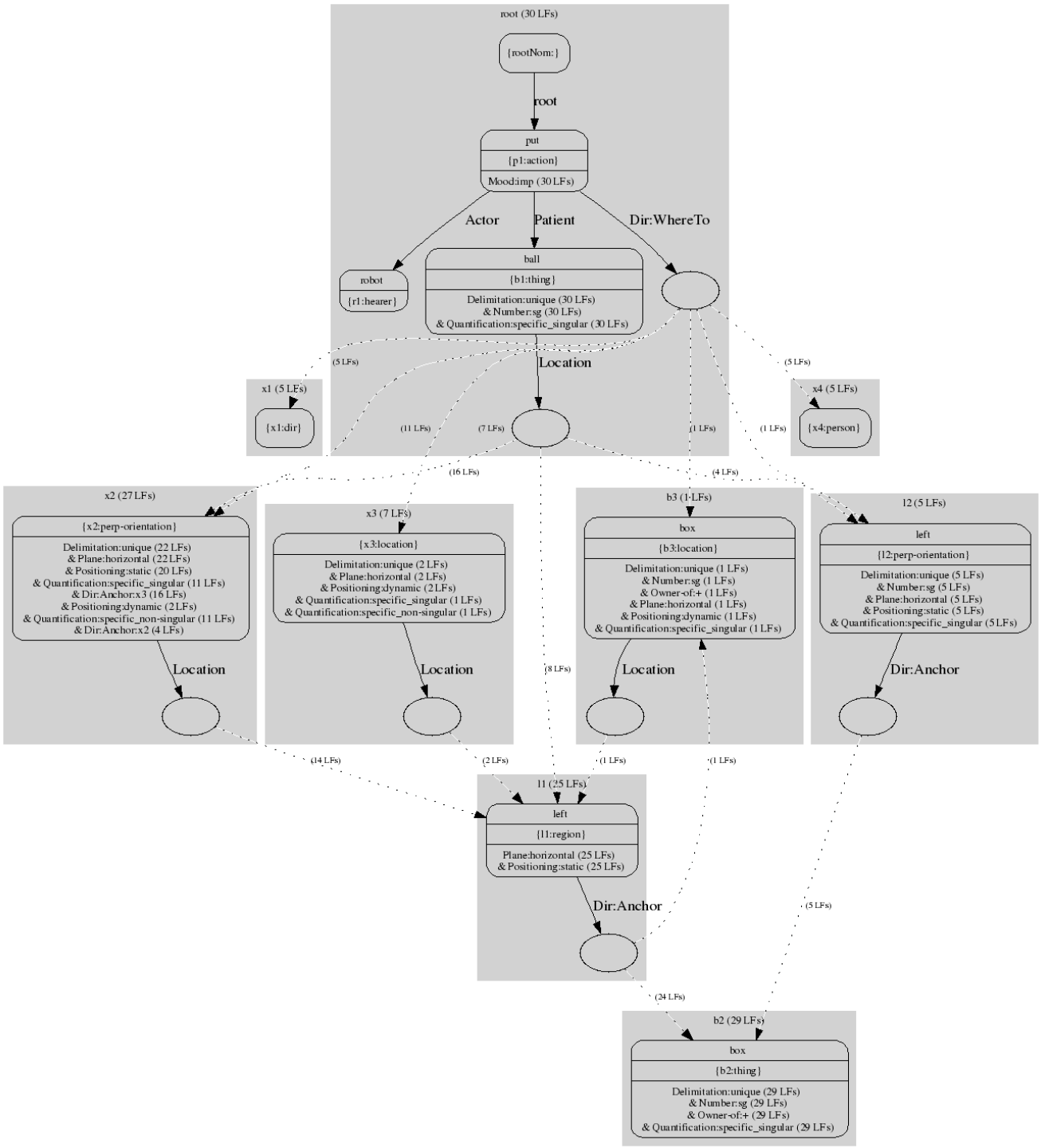


Figure 2: Example packed logical form - ”put the ball to the left of the box”

and fixates on it.

Saccade from unambiguous visual referent to next unambiguous referent, fixation: If the next expression unambiguously refers to a new visual object, the robot moves to this new object and fixates.

Saccade between ambiguous referents: If the current set of visual referents is ambiguous, the robot will move between these objects.

Saccade back to listener: Once the utterance has been fully analyzed, the robot returns to looking at the speaker.

These gaze behaviors rely in an essential way on the incremental approach to dialogue processing, as discussed above. Furthermore, a fundamental assumption is that the robot's gaze is only "natural" or believable (i.e. contributing to joint attention (Garrod and Pickering, 2004)) if it there is congruency between the intended referent and what the robot fixates on. In the next section we present the results of a pilot study which investigated this assumption.

Pilot experiment

The approach we present here is based on a fundamental assumption about the relation between gaze (notably, fixation), and visual referents. Namely, we assume that a fixation is *congruent* when the robot looks at the "right" visual referent. This has an important *hypothesized* effect on establishing joint attention in visually situated dialogue. If a robot would produce an incongruent fixation, the speaker would presumably believe that the robot did not understand her correctly. We have performed a pilot study to investigate this potential effect of incongruent fixations, in comparison to congruent fixations. This follows up on earlier studies, e.g. the one by Yoshikawa et al (Yoshikawa et al., 2006) who show that congruent gaze is perceived more natural than staring, or gaze behavior that appears independent of what the speaker communicates.

The main hypothesis for the pilot study was that "congruency between gaze fixation and intended referent leads to higher degree of belief that the robot understands which visual referent is referred to in the utterance." Or, vice versa, that incongruent gaze (i.e. fixation on the wrong visual referent) yields lower believability in the robot having understood. As baseline, we used fixation on the visually most salient item. This does not require the robot to resolve the referent. All we need to do is just trigger a behavior to look at an object. The baseline reveals how much a robot does, or does not, need to be able to relate situation awareness with dialogue processing to yield convincing interactive behavior.

The pilot study is set up as a web-experiment, in which people are shown 35 videos. Figure 4 shows a screenshot from the browser. We uploaded the videos to GoogleVideo, to make sure anyone could view them independent of platform or browser. Each video shows a visual scene of a robot with an arm, standing at a table-top scene including two or more colored objects. Some of these objects the robot is capable of manipulating, some not. Then, the robot is told an utterance, in which one visual object is referred to. While comprehending the utterance, the robot subsequently fixates at one of the objects in the visual scene. Each video takes approximately 7 to 8 seconds. After the video, the subject is asked two questions:

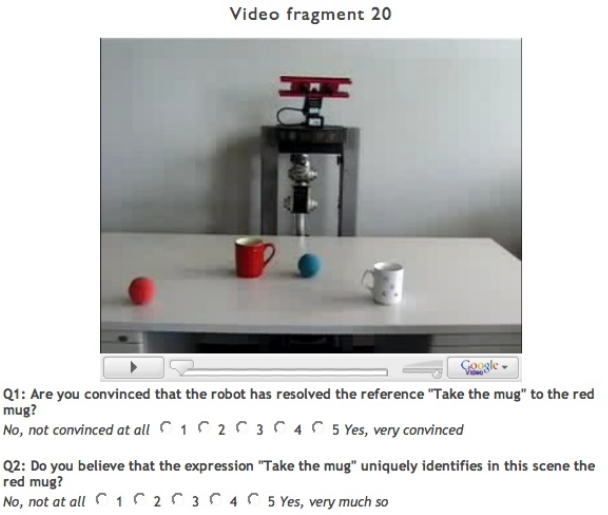


Figure 4: Browser screenshot

Q1 "Are you convinced that the robot has managed to resolve the reference XYZ to the right object (namely, Q)?" (Answer on a 5-point Lykert scale, "(1) not convinced at all ... (5) yes, totally convinced.")

Q2 "Do you believe that the expression XYZ is appropriate to uniquely identify the object Q in the scene?" (Answer on a 5-point Lykert scale, "(1) no, not appropriate at all ... (5) yes, very appropriate.")

We performed the pilot study with 15 subjects, 5 female and 10 male. Some of these subjects were familiar with robots, though none with our system. We solicited subjects by email. Subjects were not offered any financial compensation. Each subject was given the following information.

Nature of the workspace The subject is told where the robot can reach.

Nature of the objects The subject is told that all objects can be referred to as "things", which objects the robot can grasp, and that the robot can push all objects (within reach).

Nature of the instructions told to the robot The subject is told the robot may be given a command to manipulate an object, or just a description of an object in the scene

Below we discuss in more detail the principled design approach we took for generating the visual scenes for the videos, and present the results and their discussion.

Design

The point of the pilot study was to investigate congruency between gaze fixation, and visual referents. We therefore

Proc. LangRo'2007
978-972-96895-2-9

needed to design the visual scenes for the videos in such a way that we would control the factors that influence the (potential) ambiguity of a referring expression.

For each scene, we wanted to consider a number of scenarios. Given the baseline of fixating on the visually most salient object, we wanted to systematically vary the objects and scene structure relative to two fundamental conditions: (1) the intended referent has the same visual salience as a distractor, or (2), the intended referent has a lower visual salience than the visually most salient object. This implies that for a referring expression, if the visually most salient item is not in the distractor set for the expression, incongruency arises automatically (i.e. baseline gaze, versus congruent gaze).

To bring about potential (in)congruency in these conditions, we thus needed to consider the contrast between the visually most salient item, and the intended visual referent. Following (Dale and Reiter, 1995; Kelleher and Kruijff, 2006) we set up a basic template for a visual object, consisting of its material and contrastive properties, spatial relations, and visual and topokinetic salience.

Then, given a visual salience condition, we selected a set of two or more objects that would enable us to systematically vary distractor factors (relative to the intended referent) based on type, and material and contrastive properties. Subsequently, using a 4 by 5 matrix grid, we positioned objects in the scene such that we obtained the desired visual salience condition, topokinetic salience, and spatial relations that could be used to uniquely identify a referent. Figure 5 shows the salience measures (relative to the robot's viewpoint) we used in determining how these measures acted as distractor factors.

Figure 6 gives an example of a visual scene. For the condition of distinct visual salience, the utterance "Look at the ball" would have as intended referent *b*2, but would yield an incongruent gaze fixation at *m*1 under a baseline behavior (*m*1 being the visually most salient object).

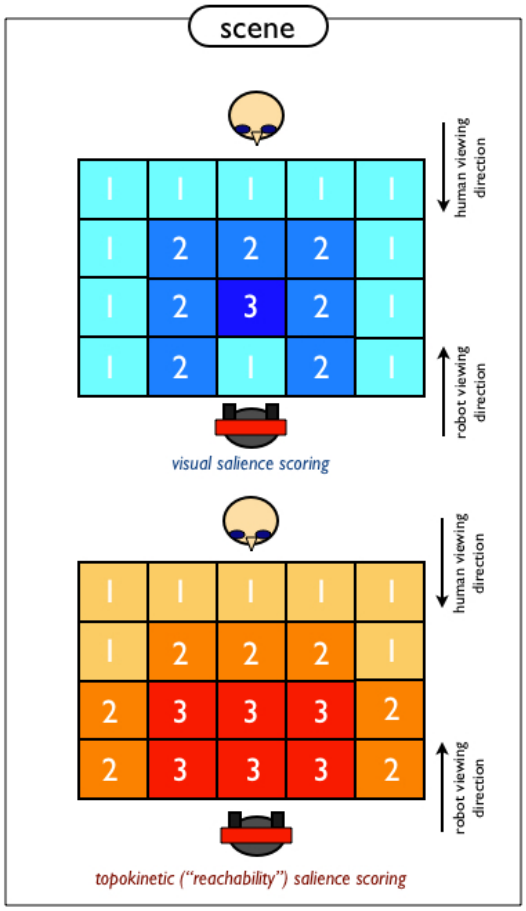


Figure 5: Visual and topokinetic salience measures

Results

We analyzed the results from the pilot study relative to the visual salience conditions. Within these conditions, we looked at two types of variance: (1) Variance in the relation between congruence/incongruence and believability (question Q1), and (2) variance in the relation between congruence/incongruence, believability (question Q1), and appropriateness of the referring expression (question Q2).

The first type of variance reveals the basic impact of (in)congruence on believability (one-way ANOVA). The second type reveals more about the relation between congruence, believability, and how much information we need in a referring expression to rule out distractors. By computing the variance in the latter type we can investigate the role visual salience plays as distractor factor.

Table 1 gives the results for the variation between congruency and believability. Results are statistically significant (p=0.001) across both conditions. Figure 7 shows the boxplot for the condition different visual salience; the

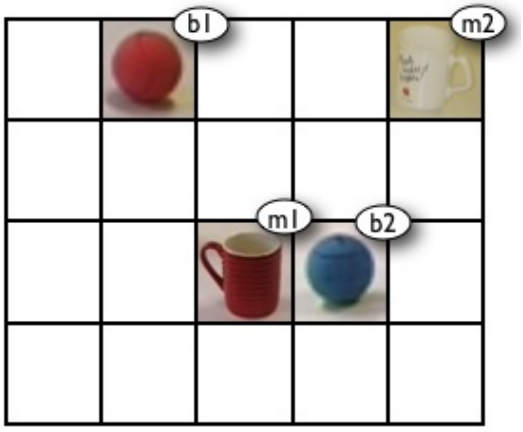


Figure 6: Sample visual scene

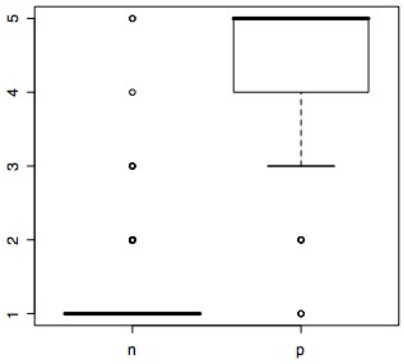


Figure 7: Boxplot: Congruency ∼ believability (diff. vis. salience)

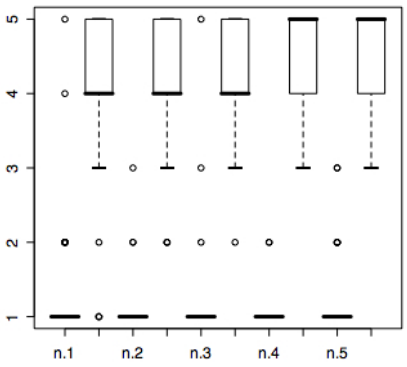


Figure 8: Boxplot: Believability ∼ congruency * appropriateness (diff. vis. salience)

boxplot for the other condition is similar.

Condition	F-value	Significance
Distractors with eq. vis. salience	961.1	0.001
Distractors with diff. vis. salience	301.08	0.001

Table 1: Variance: Congruency ∼ Believability (Q1) (one-way ANOVA)

Table 2 gives the results for the variation between congruency, believability, and appropriateness of the referring expression to uniquely identify the intended visual referent (two-way ANOVA). Again, results are statistically significant (p=0.001) across both conditions.Figure 8 and Figure 9 show the boxplots for the conditions different respectively equal visual salience.

Condition	F-value	Significance
Distractors with eq. vis. salience	299.16	0.001
Distractors with diff. vis. salience	1005.67	0.001

Table 2: Variance: Believability (Q1)∼ Congruency * Appropriateness (Q2) (two-way ANOVA)

Discussion

The results all show statistically significant interactions between congruency of fixation, and believability. The results thus confirm the main hypothesis of the pilot study.

Across the conditions, we can see an interesting pattern appear. In the condition under which visual objects are distractors because they have equal visual salience, incongruency is particularly negative (Table 1). The importance of proper linguistic reference, i.e. the production and comprehension of contextually appropriate referring expressions, because clear if we combine this result with the variance in relation to appropriateness (Table 2 and Figure 9). We need further experimentation to determine the exact impact of the different distractor factors on resolution of visual referents. Having said that, we see these results as strengthening the argument that natural language processing for human-robot interaction requires taking into

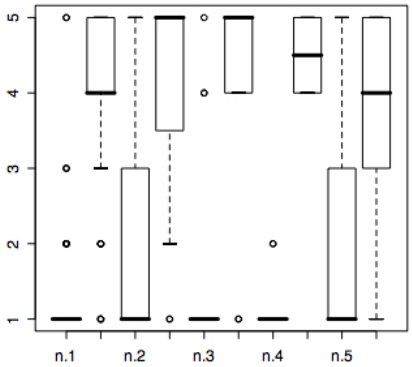


Figure 9: Boxplot: Believability ∼ congruency * appropriateness (eq. vis. salience)

account semantic and pragmatic factors – well beyond the level of simple syntactic analysis – if we want robots to produce believable gaze.

This conclusion may be strengthened further if we look at the other condition, in which distractors differ in visual salience. In this condition, incongruence arises automatically if referring expressions are not resolved. Tables 2 and 1 show the sharp contrast that can be observed between congruent and incongruent fixations in this case.

Conclusions

The paper presented an approach to robot gaze production, which we implemented using the CAST framework. The core of the approach is constituted by an incremental model of dialogue analysis, and the possibility to bind utterance meaning to visual referents. Based on what referents are becoming referred to as the utterance analysis unfolds, robot gaze is driven to move its fixation from one visual object to another. The approach is based on the assumption that, for such gaze to contribute to establishing joint attention in situated dialogue, fixations need to be congruent with the (intended) visual referents. We presented a pilot study which showed statistically significant interactions between congruence, believability, and appropriateness of referring expression – thus providing initial

Proc. LangRo'2007
978-972-96895-2-9

empirical support for the approach.

References

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001). Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–37.

Allen, J., Miller, B., Ringger, E., and Sikorski, T. (1996). A robust system for natural spoken dialogue. In *Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL'96)*, pages 62–70.

Altmann, G. and Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3):247–264.

Altmann, G. and Kamide, Y. (2004). Now you see it, now you don't: Mediating the mapping between language and the visual world. In Henderson, J. and Ferreira, F., editors, *The Interface of Language, Vision, and Action: Eye Movements and The Visual World*, pages 347–386. Psychology Press, New York NY.

Altmann, G. T. and Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Asher, N. and Lascarides, A. (2003). *Logics of Conversation*. Cambridge University Press, Cambridge, UK.

Baldridge, J. and Kruijff, G. (2002). Coupling CCG and hybrid logic dependency semantics. In *Proc. ACL 2002*, pages 319–326, Philadelphia, PA.

Baldridge, J. and Kruijff, G.-J. M. (2003). Multi-modal combinatory categorial grammmar. In *Proceedings of EACL'03*, Budapest, Hungary.

Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., Lockerd, A., and Mulanda, D. (2004a). Humanoid robots as cooperative partners for people. *Int.Jnl. Humanoid Robots*.

Breazeal, C., Hoffman, G., and Lockerd, A. (2004b). Teaching and working with robots as a collaboration. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04)*, pages 1028–1035, New York, NY.

Brenner, M., Hawes, N., Kelleher, J., and Wyatt, J. (2007). Mediating between qualitative and quantitative representations for task-orientated human-robot interaction. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*.

Brick, T. and Scheutz, M. (2007). Incremental natural language processing for HRI. In *Proceeding of the ACM/IEEE international conference on Human-Robot Interaction (HRI'07)*, pages 263 – 270.

Carroll, J. and Oepen, S. (2005). High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP'05)*, pages 165–176.

Chambers, C., Tanenhaus, M., and Magnuson, J. (2004). Actions and affordances in syntactic ambiguity resolution. *Jnl. Experimental Psychology*, 30(3):687–696.

Dale, R. and Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

DeVault, D. and Stone, M. (2003). Domain inference in incremental interpretation. In *Procedings of the Fourth Workshop on Inference in Computational Semantics (ICOS'04)*.

Garrod, S. and Pickering, M. (2004). Why is conversation so easy? *Trends in Cognitive Sciences*, 8:8–11.

Gorniak, P. and Roy, D. (2004). Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470.

Gorniak, P. and Roy, D. (2005). Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005)*.

Gorniak, P. and Roy, D. (2007). Situated language understanding as filtering perceived affordances. *Cognitive Science*, 31(2):197–231.

Hawes, N., Sloman, A., Wyatt, J., Zillich, M., Jacobsson, H., Kruijff, G., Brenner, M., Berginc, G., and Skocaj, D. (2007a). Towards an integrated robot with multiple cognitive functions. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI'07)*.

Hawes, N., Zillich, M., and Wyatt, J. (2007b). BALT & CAST: Middleware for cognitive robotics. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, pages 998 – 1003.

Kelleher, J. (2005). Integrating visual and linguistic salience for reference resolution. In *Proceedings of the 16th Irish conference on Artificial Intelligence and Cognitive Science (AICS-05)*.

Kelleher, J. and Kruijff, G. (2006). Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1041–1048.

Knoeferle, P. and Crocker, M. (2006). The coordinated interplay of scene, utterance, and world knowledge: evidence from eye tracking. *Cognitive Science*.

Kruijff, G., Kelleher, J., and Hawes, N. (2006). Information fusion for visual reference resolution in dynamic situated dialogue. In André, E., Dybkjaer, L., Minker, W., Neumann, H., and Weber, M., editors, *Perception and Interactive Technologies (PIT 2006)*. Spring Verlag.

Kruijff, G., Zender, H., Jensfelt, P., and Christensen, H. (2007). Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems, Special Issue on Human and Robot Interactive Communication*, 4(1):125–138.

Miyauchi, D., Sakurai, A., Makamura, A., and Kuno, Y. (2004). Active eye contact for human-robot communication. In *Proceedings of CHI 2004*, pages 1099–1104. ACM Press.

Mori, D., Matsubara, S., and Inagaki, Y. (2001). Incremental parsing for interactive natural language interface. In *2001 IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 2880–2885.

Oepen, S. and Carroll, J. (2000). Ambiguity packing in constraint-based parsing: Practical results. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000)*, pages 162–169.

Pickering, M. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–225.

Rosé, C. P., Roque, A., Bhembe, D., and VanLehn, K. (2002). An efficient incremental architecture for robust interpretation. In *Proceedings of the Human Languages Technologies Conference*.

Scheutz, M., Eberhard, K., and Andronache, V. (2004). A real-time robotic model of human reference resolution using visual constraints. *Connection Science Journal*, 16(3):145–167.

Sidner, C. L., Kidd, C. D., Lee, C. H., and Lesh, N. (2004). Where to look: A study of human-robot engagement. In *ACM International Conference on Intelligent User Interfaces (IUI)*, pages 78–84. ACM.

Sidner, C. L., Lee, C., Kidd, C., Lesh, N., and Rich, C. (2005). Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2):140–164.

Sikkel, K. (1999). *Parsing Schemata*. Springer Verlag.

Steedman, M. (2000). *The Syntactic Process*. The MIT Press, Cambridge MA.

Stone, M. and Doran, C. (1997). Sentence planning as description using tree-adjoining grammar. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, pages 198–205.

Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., and Sedivy, J. (1994). Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.

Yoshikawa, Y., Shinozawa, K., Ishiguro, H., Hagita, N., and Miyamoto, T. (2006). Responsive robot gaze to interaction partner. In *Proceedings of Robotics: Science and Systems II (RSS'06)*, pages 287–294.

Towards Speech-Based Human-Robot Interaction

Roger K. Moore

Dept. Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK
r.k.moore@dcs.shef.ac.uk

Abstract

Notwithstanding the success of contemporary spoken language technology in a range of practical applications, it is widely acknowledged that serious shortfalls in performance limit its wider deployment. Unconstrained speech-based interaction with embodied agents - such as robots - remains outside the scope of current technology and thus presents key challenges to the research community. However, it is argued that the solutions lie, not only outside the field of speech technology, but also outside current theories of human spoken language processing. Instead, it is proposed that research into spoken language by mind *or* machine now needs to draw inspiration from areas as widely dispersed as cognitive neuroscience and control engineering. Following such an approach, this paper describes a theoretical framework known as 'PREdictive SENsorimotor Control and Emulation' (PRESENCE), and experiments using a PRESENCE-inspired architecture to enable a robot to clap in synchrony with a user's voice illustrate the power of the paradigm. It is concluded that future research in spoken language processing is likely to benefit greatly from PRESENCE and from greater emphasis on the challenges raised in situated and embodied environments, the evolution and acquisition of spoken language, and appropriate and intuitive speech-based human-robot interaction.

Introduction

Over the past fifty years, spoken language technology – automatic speech recognition, text-to-speech synthesis and spoken language dialogue systems – has made tremendous strides in terms of its technical abilities and practical applications. The majority of mobile telephones now carry 'voice dialling' as a standard feature, the new Microsoft Vista operating system incorporates the ability to dictate documents or control a PC by voice, and IVR (interactive voice response) systems are becoming commonplace for interacting with automated services over the telephone. Progress has been driven by the extensive use of machine learning techniques drawing on vast quantities of speech training material.

However, these successes belie the uncomfortable fact that the performance of such systems appears to be asymptoting well short of human spoken language capabilities, and such shortfalls reveal themselves in realistic everyday environments which may contain competing sound sources, multiple users or which inadvertently encourages users to step outside the narrow confines of the application domain. Unfortunately each of these aspects typifies the range of applications that involve speech-based interaction with embodied agents - such as robots - and hence the feasibility of integrating contemporary spoken language technology into robotic systems is currently severely compromised.

Nevertheless, the challenges posed by attempting to speech-enable robotic systems are exactly those that can drive spoken language technology research in fruitful new directions. The author has argued elsewhere (Moore, 2007a) that the limitations of current spoken language technology are a direct consequence of the natural tendency of scientists to take a reductionist approach in which automatic speech recognition, synthesis and dialogue are treated as independent components and even developed by different research communities. Such enforced separation also undermines those few attempts that have been made to 'bridge the gap' between automatic and human speech processing (Scharenborg et al, 2003).

The Way Forward

What appears to be needed to move to the next generation of spoken language technology is to re-evaluate the current research paradigms not, as one might suppose, with respect to current theories of human spoken language (which are similarly fragmented), but in the light of a number of advanced ideas drawn from disciplines *outside* the field of spoken language processing. In particular, considerable progress is currently being made (in areas such as cognitive neuroscience) in understanding and modelling the general behaviour of living systems, and much of this research is directly relevant to spoken language interaction. Old ideas such as 'perceptual control theory' (Powers, 1973) and new discoveries such as 'mirror neurons' (Rizzolatti and Craighero, 2004) serve to indicate a hitherto unsuspected and intimate link between perceptual and productive behaviours and inspire new models of action understanding based on significant sensorimotor overlap. Coupled with contemporary theories of cortical functionality such as 'hierarchical temporal memory' (Hawkins, 2004) and 'emulators' (Grush, 2004), these putative processes offer a tantalising glimpse into possible computational models of cognition, interaction and speech.

Predictive Sensorimotor Control and Emulation

In (Moore, 2007a and 2007b), the author has drawn a number of such ideas together into a single coherent theoretical framework termed PRESENCE – 'PREdictive SENsorimotor Control and Emulation' - a core feature of which is the necessity to move away from a classic

Brunswikian stimulus-response model of behaviour to one in which participants (humans or machines) are viewed as multiple loosely-coupled control-feedback loops. It is argued that such an approach provides a more sophisticated model of interactive behaviour such as spoken language and provides a putative architecture for future speech-based human-machine interaction in situated embodied environments.

PRESENCE is based on the premise that there are three fundamental factors that ultimately determine an organism's fitness to survive in an evolutionary framework: its ability to manage **energy** (facilitating efficient behaviour in the context of scarce resources), **time** (facilitating efficient planning in the context of potentially harmful situations) and **entropy** (facilitating efficient communications in the context of information sparsity). These constraints, coupled with an integrated and recursive processing architecture, pave the way to a new approach to spoken language technology in which high-level interactive behaviours such as prosody and emotion emerge as essential aspects of a communicative system rather than as processing afterthoughts.

Experimental Work

A preliminary experimental validation of the principles espoused in PRESENCE has been conducted using the ALPHA REX humanoid robot constructed using the LEGO® MINDSTORMS® NXT platform. By coordination and synchronization in a PRESENCE-inspired framework, the robot was able to learn to produce motor behaviour in time to rhythmic spoken input (much like someone clapping along to music).

The robot was programmed using three sensorimotor control loops: one to monitor and control its own behaviour, one to monitor the behaviour of the human user and a third driven by a 'need' to optimise synchronisation between the other two. The resulting behaviour is illustrated in figure 1.

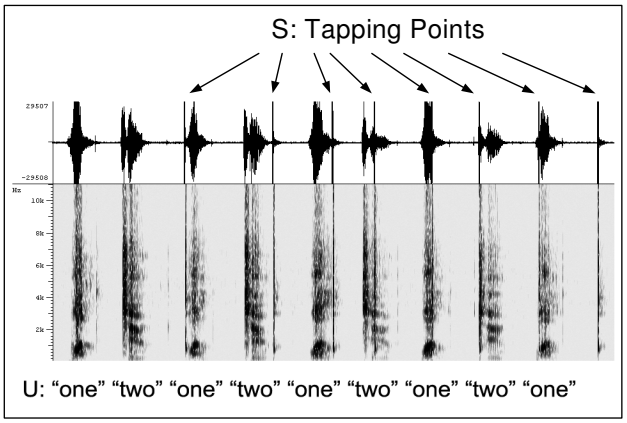


Fig.1: Robot tapping in synchrony with a user's speech (S-system, U-user).

The results of the experiment showed that the robot was not only able to synchronise its behaviour with that of the user, but it also successfully predicted successive rhythmic actions after the user ceased to speak.

Conclusion & Future Research

As a result of the development of PRESENCE and the preliminary experiments reported here, it is concluded that future research in spoken language processing is likely to benefit greatly from greater emphasis on the very practical issues raised in situated and embodied environments, and the computational mechanisms required to support appropriate and intuitive speech-based human-robot interaction. To that end, research at the University of Sheffield is currently being directed towards models of the evolution and acquisition of spoken language (Boves et al, 2007), and the development of an animatronic tongue - see figure 2.

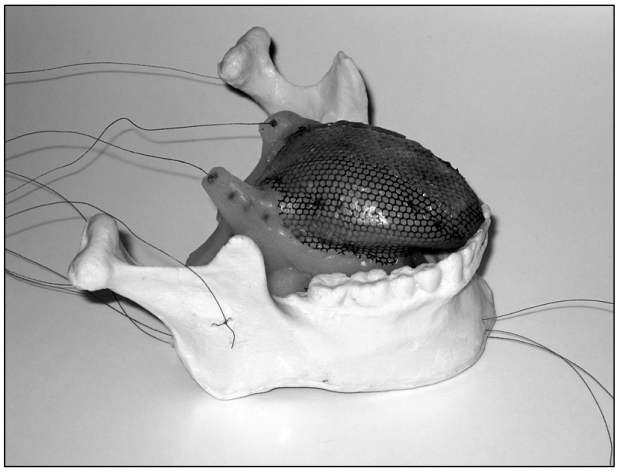


Fig. 2: Animatronic tongue being developed at the University of Sheffield.

References

Boves L., ten Bosch L. and Moore R.K. 2007. ACORNS: Towards computational modeling of communication and recognition skills, Proc. 6th IEEE Int. Conf. on Cognitive Informatics, Lake Tahoe, CA, USA.

Grush, R. 2004. The emulation theory of representation: motor control, imagery, and perception, Behavioral and Brain Sciences 27:377-442.

Hawkins, J. 2004. On Intelligence, Times Books.

Moore, R.K. 2007a. Spoken language processing: piecing together the puzzle, Speech Communication 49:418-435.

Moore, R.K. 2007b. PRESENCE: A human-inspired architecture for speech-based human-machine interaction, IEEE Trans. Computers, 56:1176-1188.

Powers, W.T. 1973. Behaviour: The Control of Perception, Hawthorne, NY: Aldine.

Rizzolatti, G. and Craighero, L. 2004. The mirror-neuron system, Annual Review of Neuroscience 27:169-192.

Scharenborg, O., ten Bosch, L., Boves, L. and Norris, D. 2003. Bridging automatic speech recognition and psycholinguistics: extending Shortlist to an end-to-end model of human speech recognition, J. Acoustical Soc. of America 114(6):3023-3035.

Constructing an Interactive Behavioral On-line Operation System for Humanoid Robots Using Natural Language Instructions

Ee Sian Neo, Takeshi Sakaguchi and Kazuhito Yokoi
National Institute of Advanced Industrial Science and Technology,
Tsukuba, Ibraki 305-8568, Japan.
rio.neo@aist.go.jp

Abstract

In the aim of realizing user-friendly humanoid robots that are flexible, adaptable and easy to command and instruct, this paper reports on our endeavor to construct an on-line operation system that enables a human user to operate humanoid robots by using natural language instructions.

To realize artificial agents that are able to understand natural language instructions and act accordingly will need the integration of knowledge representation, perception, decision-making and on-line motion generation technologies. Humanoid robots, which are mobile platforms that possess multi-degrees of freedom and multi operational point, will need to address unique challenges apart from the general difficulties of constructing natural language interpretation system.

Our research group has been developing on-line behavioral operation technologies that enable humanoid robots to perform tasks in human environment integrating object recognition technologies using 3D vision and online whole-body motion generation technologies. This paper tackles this integration problem by addressing the issues of representing knowledge of objects and actions which facilitates natural language instructions for tasks in indoor human environments. We propose a taxonomy of objects in indoor human environments and a lexicon of basic actions and behaviors in this preliminary attempt to construct a reliable and flexible natural language instruction system.

We describe the implementation of the proposed on-line behavioral operation system on our humanoid robot HRP-2, which is able to detect the direction of a speaker from within 2 meters and receive natural language instructions from the user using microphone arrays connected to a speech recognition embedded system on-board the robot(Figure1).

For an autonomous humanoid robot, as its control module, information processing module, batteries, and mechanical components have to be built on-board the robot, the number and performance of CPU that can be installed on-board the robot is limited. For this reason, while visual processing are usually processed on-board of the robot due to the difficulty to have high-bandwidth communication for high quality on-line visual information, speech information is relatively low and its recognition are usually done remotely. In our system, we have adopted a

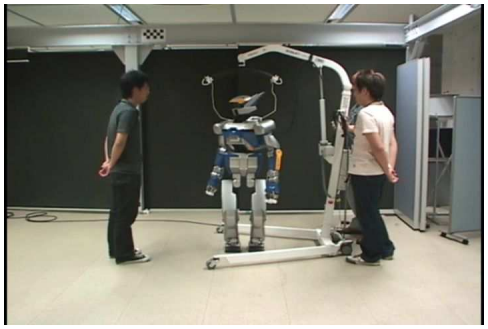


Figure 1: Responding to speaker: Yes, can I help you?.

Figure 2: Taking a drink from the fridge as instructed.

speech recognition hardware module developed by NEC Corporation in our joint project, the Development Project for a Common Basis of Next-Generation Robots (Development of Speech Recognition Device and Module) sponsored by the New Energy and Industrial Technology Development Organization of Japan. The speech recognition hardware module is developed using an application processor MP211 manufactured by NEC Electronics which has three ARM9 CPUs and one DSP mounted on an embedded system on-board our robot HRP-2. This speech recognition module is able to process the speech information input from an 8-channel microphone array on-board HRP-2 for speaker direction detection and speech recognition with noise filtering.

We will report on experiments using the proposed online behavioral operation system which enable a human operator to instruct HRP-2 to perform tasks such

as taking a drink from the fridge using a combination of basic actions and behaviors through natural language instructions(Figure2,Figure3).

Figure 3: Giving the requested drink during demonstration.

Proc. LangRo'2007
978-972-96895-2-9

Meaning in the timing?
The emergence of complex pointing patterns

Katharina J. Rohlfing[1] and Stefan Kopp[2]

Bielefeld University
[1]rohlfing, [2]skopp@techfak.uni-bielefeld.de

Abstract

We present observations that suggest that pointing can fulfill not only a deictic but also an iconic function. We discuss the possibility that the iconic patterns emerge mainly because of tight timing and close sequence of a deictic gestures and another, iconic gesture. Thus, instead of overloading single gesture performances with deictic and iconic information at the same time, complex pointing patterns is hypothesized to emerge from two distinct gestures that convey meanings so tightly related that the movements are triggered and generated either closely one after the other or even in overlap.

We hypothesize that effects due to the biological constraints of the human body and co-articulation as well as the time characteristics involved in this form of incremental gesture thinking may have a significant impact on the resulting gesture pattern. This performance may, in turn, be perceived and interpreted by the observer as complex in terms of different forms of information. We tested this alternative hypothesis by analyzing empirical data on pointing in mother-child dialogues. In addition, we simulated the emergence of pointing patterns using an artificial system, the virtual human MAX, whose motor control model is able to fluently blend independent, successive gestures that can be exactly timed.

Motivation

In this paper, we argue that the timing in-between communicative gestures can convey specific meaning either by itself or because a timed information is meaningfully perceived. Imagine a child acquiring first language: When she or he says "door!" and then shortly after "open!", this behavior can be interpreted either as two utterances, e.g. that the child wants to open the door or as a two-word-sentence, e.g. by which the child expresses that the door is open. Certainly, it depends on the situation which interpretation is more appropriate but it is also a matter of timing: When the time between the two utterances is very tight, it is more likely to be a two-word-sentence. We would like to extend this notion to gestural behavior which is assumed to be tightly coupled with thinking for speaking, but which adds to it the dimension of spatio-motoric imagery.

Introduction

In studies about gestural behavior, the timing of the performed gestures and the uttered speech is crucial. In fact, gestures and speech are viewed to be semantically and pragmatically co-expressive (Butcher & Goldin-Meadow, 2000). This implies that the production of verbal and nonverbal behavior is synchronous. Some investigators let a time period of 2 seconds decide about the relationship between gesture and speech. Other decide that one concept in speech has to be expressed by one concept in gesture (Gerwing, 2007). However, in praxis, kinetic information alone seems to be insufficient for spotting a gesture (Sowa, 2007). Therefore, a researcher analyzing gestural behavior is confronted with various cases in which (1) it is difficult to decide whether a particular gesture stands in relation with what has been said and (2) a gesture appears to be more complex, i.e. composed of some units. For example, when mothers explain to their children how to bring two objects in a certain spatial relation, they point not only to the participating objects but they also indicate which motion is necessary to perform this relation: Their point almost seems to draw a trajectory according to which a child should put the objects together. Similarly, when pointing to explain a relation that is considered canonical between two objects (e.g. a train *inside* a tunnel), mothers seemed to produce pointing gesture in form of a saccade, i.e. from the trajector to the landmark object. It was hypothesized that such a saccade may convey additional semantic information about the togetherness (canonicality) of the two objects (Rohlfing 2005).

Together, these observations suggest that a pointing gesture can fulfill not only a deictic but also an iconic function. This would support the recent view that the gesture types often proposed in the literature are not disjunctive but constitute different dimensions which may apply to different extents to the same gesture (McNeill 2005). However, an alternative possibility is that pointing retains its primarily deictic meaning but the observed patterns emerge mainly from the tight timing and close sequence of one pointing gesture and another, iconic gesture. Thus, two distinct gestures would be responsible for the appearance of a pointing pattern mainly because of the dynamics and incrementality of the verbal-gestural thinking that has brought them to life. For example, an observed saccade might not be intentionally drawn but come about from the necessary motor motions when performing fast and sequential pointing to different participants. Similarly, the motion trajectory could emerge because the speaker starts with a deictic gesture, isolating the location of an intended referent, and then focuses her thinking on closely related, yet different (iconic) information that is reflected in a seamlessly connected iconic gesture, retaining the previously adopted hand shape.

Thus, instead of overloading single gesture performances with deictic and iconic information at the same time, complex pointing patterns are hypothesized to emerge from two or more distinct, successive gestures that convey meanings so tightly related that the movements are triggered and generated either closely one after the other or even in overlap. The final gestural performance will result not only from the individual form of every single gesture, but also from how they can be combined and reconciled in accord with the gesture-specific timing constraints. We hypothesize that effects due to the biological constraints of the human body and motor control, i.e. co-articulation, as well as the time characteristics of the processes involved in this form of incremental gesture thinking may have a significant impact on the resulting gesture pattern. This performance may, in turn, be perceived and interpreted by the observer as complex in terms of different forms of information.

Study

We tested our alternative hypothesis by analyzing empirical data on pointing in mother-child dialogues (Rohlfing, 2005). In addition, we simulated the emergence of pointing patterns using an artificial system, the virtual human MAX, whose motor control model allows us to exactly define the timing of successive gestures and then creates fluent blending and co-articulation in-between them automatically (Kopp and Wachsmuth 2004).

Participants. 21 students from the Bielefeld University were recruited for this study.

Stimuli. The participants were presented four scenes that were selected from a corpus of 34 mother-child pairs and five scenes that were created with the virtual human MAX. For the human data, the scenes were examples of a natural mother-child conversation, in which a mother produced gestural behavior according to the categories below. The scene was cut out of the context of a task, in which a mother was asked to instruct her child to put to objects together.

For MAX data, the scenes were re-constructed by manually defining the utterances to be produced by the agent in an XML specification language. Such specifications state the words to be uttered by the system, the form of the gestures, and the words in speech with which a gesture is affiliated (and must thus be synchronized with).

Two scenes were chosen (and reconstructed with MAX) to present single pointing:

- a pointing gesture
- an iconic gesture

Three other examples were chosen to present a complex pointing behavior:

- a saccade leading from one object to another (see Figure 1).

Figure 1: A pointing saccade in natural mother-child conversation

In MAX, this was modeled as two separate, but tightly timed pointing gestures, for which the system would generate a shortest-path, linear transition movement in-between (see Figure 2).

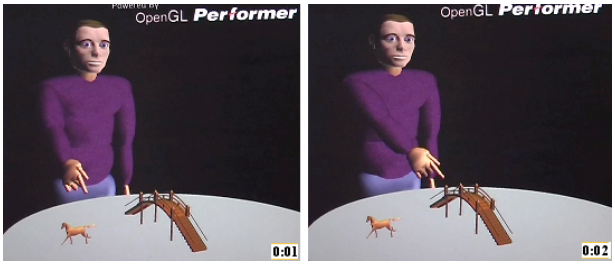


Figure 2: A pointing saccade in MAX

- a single trajectory going from one object to another with more iconicity in the transition (see Figure 3).

Figure 3: An "expressive transition" in mother-child conversation

In MAX, the performance was modeled as two separate gestures (see Figure 4); the second gesture required a change of gesture direction, from which more depictive movements (thus potential iconicity) in the motor transition emerged — we called it "expressive transitions"

Proc. LangRo'2007
978-972-96895-2-9

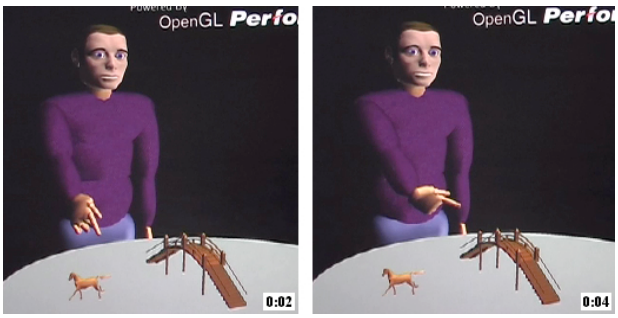


Figure 4: An "expressive transition" in MAX

• a similar transition as above, however, instead of two separate gestures, MAX performed a dynamic gesture with pointing hand shape, with the preparation phase towards one object (at 0:02) and then the stroke of the gesture turning towards the other object (at 0:03) — we called this pattern a "transitive expression".

The participants were randomly assigned to two groups: One group saw gestures accompanying speech, in the other group, only the gestural behavior (without the audio) was presented.

Procedure. All the scenes were presented in a random order. The participants were invited to rate the gestures that they see on a computer screen. After we gave some examples, we asked them to decide whether they see a deictic or iconic gesture, how many gestures they see, and what the gestures refer to (an object, an action, or a location).

Results. The single pointing gestures were rated similarly for the scenes from mother-child conversations and for MAX, i.e. no statistical differences between the scenes were found: The majority of subjects saw one gesture in the presentation.

As to the complex pointing patterns, the participants described the pointing saccade as two separate pointing gestures in both scenarios.

	Scene from human data	Scene with MAX
1 gesture	24	9.5
2 gestures	76	90.5
deictic	86	95

Table 1: Subjects' ratings (in percentage) for a pointing saccade

As shown in Table 1, even though no statistical differences were found between the human data and MAX (paired t-test $t(20) = -1.4$, $p = 0.19$ concerning the number of gestures seen), the ratings in human data seems to be less obvious than for the MAX scene. This might result from the fact that in MAX, the two gestures were generated separately and necessarily with lesser continuity as in the human data. We can therefore conclude that a pattern does not emerge when the transition has little iconicity and/or little continuity.

As to the "expressive transitions", our analysis revealed that while in the human data, subjects' decision about the number of gesture was quite uncertain (43 % identified one, 57 % two separate pointing gestures), in the MAX data, subjects identified rather two pointing gestures (86 %). We think that for human data, more examples have to be presented in order to investigate whether a change in directionality of the second gesture will be perceived as a separate gesture, which seems to be the case in the MAX data.

Interestingly, when the gesture is programmed as a single gesture that prepares with pointing to one object and whose stroke reorients itself into a differently directed pointing to another object ("transitive expression"), then it seems to be a matter of the accompanying speech how this gestures is perceived (see Table 2).

	With speech	No speech
1 gesture	36	70
2 gestures	64	30
deictic	91	40

Table 2: Subjects' ratings (in percentage) for a "transitive expression"

In the group who heard the speech along with the gesture (left column in Table 2), subjects were inclined to identify two deictic gestures. In the group without audio (right column in Table 2), subjects identified rather a single iconic gesture. These group differences in identifying the kind of pointing pattern were statistically significant as shown by a paired t-test: $t(20) = 2.79$, $p < 0.05$. That is, our initial hypothesis that when the iconicity in the transition is increased, subjects will identify one rather then two gestures proved to be too simple. We only obtained support for it in conditions in which only the mere gestural movements are available to the raters. When additional information from speech comes into play, subjects tended to see two gestures, even though only one continuous movement was performed by MAX $r(19) = 0.67$, $p < 0.01$. Thus, even though the gestural pattern seems to visually indicate a single meaningful motor performance, as suggested by the results from the no-speech condition, the subjects' rating is heavily influenced by what they heard the system saying: *put the [**horse** under the bridge]!* It is possible that the impression of two gestures being present emerged from the idea that two objects are involved, which were marked or 'pointed out' by the preparation phase coinciding with the word "horse" and the stroke coinciding with the word "bridge".

Conclusions

Our study was the first one that explicitly compared gestures that were performed by real speakers with gestures that were autonomously rendered by an artificial system. The parallels we found between these two gesture sets suggest that complex pointing behavior can be modeled in an artificial system like the virtual human MAX, and that this method may be viable as a means of systematically exploring variants of verbal or nonverbal communicative behavior. In applying this method here, we could isolate novel effects of the verbal utterance on the perception and interpretation of pointing patters, instances of everyday gestural behavior that is so prominent and natural as to be frequently used even by mothers when they explain to the children basic spatial relations.

In our future work, we aim to deepen both our research method as well as the study described here. For the latter, we plan to provide information about the performed task to the subjects in order to achieve a more context sensitive rating. In addition, we plan to develop settings for adult-adult communication and aim to elicit pointing behavior in order to gain more examples for our fine-grained analysis of reasons for the emergence of complex pointing patterns.

Acknowledgment

The research reported in this paper was made possible by the Dilthey Fellowship (Research initiative Focus on the Humanities) of the VolkswagenStiftung to Katharina Rohlfing. We would like to express our gratefulness to our Research Assistants for their work: Kerstin Kickert and Angela Grimminger for the organization and administration of the study.

References

Butcher, C. and Goldin-Meadow, S. 2000. Gesture and the transition from one- to two-word speech: when hand and mouth come together. In D. McNeil ed. Language and gesture. Cambridge: CUP: 235-257.

Gerwing, J. 2007. A social approach to gesture use. Paper presented at the 3d International Conference of the International Society for Gesture Studies, Evanston, USA, 18. – 21. Juni.

Kopp, S. and Wachsmuth, I. 2004. Synthesizing Multimodal Utterances for Conversational Agents. Computer Animation & Virtual Worlds 15(1): 39-52.
McNeill, D. 2005. Gesture & Thought. Chicago: University of Chicago Press.

Rohlfing, K. J. 2005. Pointing to spatial relations in mother-child dialogue. In Proceedings of the 5th Workshop on Language and Space (Workshop on Spatial Language and Dialogue).

Sowa, T. 2007. the influence of visual context on gesture perception. Paper presented at the 3d International Conference of the International Society for Gesture Studies, Evanston, USA, 18. – 21. Juni.

Scaling Up Category Learning for Language Acquisition in Human-Robot Interaction

Luís Seabra Lopes[1,2] and Aneesh Chauhan[2]

[1]Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Portugal
[2]Actividade Transversal em Robótica Inteligente, IEETA, Universidade de Aveiro, Portugal

{ lsl, aneesh.chauhan } @ ua.pt

Abstract

Motivated by the need to support language-based communication between robots and their human users, as well as grounded symbolic reasoning, this paper presents a learning architecture that can be used by robotic agents for long-term and open-ended category acquisition. In this learning architecture, multiple object representations and multiple classifiers and classifier combinations are used. All learning computations are carried out during the normal execution of the agent, which allows continuous monitoring of the performance of the different classifiers. The measured classification successes of the individual classifiers support an attentional selection mechanism, through which classifier combinations are dynamically reconfigured and a specific classifier is chosen to predict the category of a new unseen object. In the current implementation of this learning architecture, base classifiers follow a memory-based approach, in which misclassified instances are simply added to the instance database. The main similarity measures used in the implementation are based on Euclidean distance and on a multi-resolution matching algorithm. Classifier combinations are based on majority voting and the Dempster-Shafer evidence theory. A simple agent, incorporating these learning capabilities, is used to test the approach. A long-term experiment was carried out having in mind the open-ended nature of category learning. With the help of a human mediator, the agent incrementally learned 68 categories of real world objects visually perceivable through an inexpensive camera.

Introduction

Human-robot interaction is currently a very active research field (Fong et al. 2003). The role of social interaction in machine learning and, particularly, in robot learning is being increasingly investigated (Seabra Lopes and Connell 2001; Thomaz and Breazeal, 2006).

Robots are expected to adapt to the non-expert user. This adaptation includes the capacity to take a high-level description of the assigned task and carry out the necessary reasoning steps to determine exactly what must be done. Adapting to the user also implies using the communication modalities of the human user. Spoken language is probably the most powerful communication modality. It can reduce the problem of assigning a task to the robot to a simple sentence, and it can also play a major role in teaching the robot new facts and behaviors. There is, therefore, a trend to develop robots with spoken language capabilities (Seabra Lopes and Connell 2001; Steels and Kaplan 2002; Fong et al. 2003; Seabra Lopes et al. 2005).

Language processing, like reasoning capabilities, involves the manipulation of symbols. By symbol it is meant a pattern that represents some entity in the world by association, resemblance or convention (Seabra Lopes and Chauhan 2007). Association and resemblance arise from perceptual, sensorimotor and functional aspects while convention is socially or culturally established. In classical artificial intelligence, symbolic representations were amodal in the sense that they had no obvious correspondence or resemblance to their referents Harnad (1990) proposed a hybrid approach to the "symbol grounding problem," which consisted of grounding bottom-up symbolic representations in iconic representations and categorical representations.

A distributed view on language origins, evolution and acquisition is emerging in linguistics. This trend emphasizes that language is a cultural product, perpetually open-ended and incomplete, ambiguous to some extent and, therefore, not a code (Love 2004). The study of language origins and evolution has been performed using multi-robot models, with the Talking Heads experiments as a notable example (Steels 1999; Steels 2001). In this case language is transmitted horizontally in the population of robots. Meanwhile, processes where language is vertically transmitted are of particular relevance to robotics applications. In vertical transmission, an agent or population of agents inherits most of its linguistic behavior from a previous generation, or from an independent population (Steels 2003; Steels and Kaplan 2002). Given that language acquisition and evolution, both in human and artificial agents, involve not only internal, but also cultural, social and affective processes, the underlying mechanism has been called "external symbol grounding" (Cowley 2007).

Having in mind the need to support symbolic reasoning and communication mechanisms in artificial agents, this paper investigates how category learning can be implemented in such agents. As in other works reported in the literature, this topic will be explored in a visual category learning domain. Such popular choice is justified by analogies with child development. In fact, in the earliest stages of child language development, most of the vocabulary consists of common nouns that name

concrete objects in the child's environment, such as food, toys and clothes. Gillette et al. (1999) show that, the more imageable or concrete the referent of a word is, the easier it is to learn. So concrete nouns are easier to learn than most verbs, but "observable" verbs can be easier to learn than abstract nouns.

Cognitive models and robotic prototypes have been developed for the acquisition of a series of words or labels for naming certain categories of objects. In general, the success of language acquisition in robots depends on a number of factors (Seabra Lopes and Chauhan, 2007): sensors; active sensing; physical interaction with objects; consideration of the affordances of objects; interaction with the human user; object and category representations; category learning; category membership evaluation. Most of these issues still need to be suitably addressed by robotics researchers.

Roy and Pentland (2002) present a system that learns to segment words out of continuous speech from a caregiver while associating these words with co-occurring visual categories. The implementation assumes that caregivers tend to repeat words referring to salient objects in the environment. Therefore, the system searches for recurring words in similar visual contexts. Word meanings for seven object classes were learned (e.g., a few toy animals, a ball). Steels and Kaplan (2002) use the notion of "language game" to develop a social learning framework through which an AIBO robot can learn its first words with human mediation. The mediator, as a teacher, points to objects and provides their names. Names were learned for three objects: "Poo-Chi," "Red Ball" and "Smiley." The authors emphasize that social interaction must be used to help the learner focus on what needs to be learned. Yu (2005) studies, through a computational model, the interaction between lexical acquisition and object categorization. In a pre-linguistic phase, shape, color and texture information from vision is used to ground word meanings. In a later phase, linguistic labels are used as an additional teaching signal that enhances object categorization. A total of 12 object categories (pictures of animals in a book for small children) were learned in experiments.

The authors of the present paper have previously developed a vocabulary acquisition and category learning system that integrates the user as instructor (Seabra Lopes and Chauhan 2006; Seabra Lopes and Chauhan 2007). The user can provide the names of objects as well as corrective feedback. An evaluation methodology, devised having in mind the open-ended nature of word learning, was proposed and used. On independent experiments, the system was able to learn 6 to 12 categories of regular office objects, associating them to natural language words. Like us, Lovett et al. (2007) also advocate that the key to recognition in the absence of domain expectations (i.e. in open-ended domains) is efficient on-line learning, but the work they describe is still based on the traditional procedures of gathering instances manually, training a recognizer on some of them and finally testing on unseen instances. The most notable feature of this work is the use of qualitative image representations and a specific similarity assessment method. The approach is demonstrated by learning 8 categories of user-drawn sketches. Another recent work also explores continuous learning for visual concepts (Skocaj et al, 2007). They use very simple objects to teach four colour categories (red, green, blue, yellow), two size categories (small, large) and four shape categories (square, circular, triangular, rectangular).

Current approaches to the problem, although quite different from each other, all seem to be limited in the number of categories that can be learned (usually not more than 12 categories). This limitation seems also to affect incremental/lifelong learning systems, not specifically developed for word learning or symbol grounding, such as Learn++ (Polikar, Udpa, Udpa & Honavar 2001) and EBNN (Thrun 1996). Several authors have pointed out the need for scaling up the number of acquired categories in language acquisition and symbol grounding systems (Cangelosi and Harnad 2000; Steels and Kaplan 2002).

Within the field of computer vision, there is recent progress towards systems able to learn larger numbers of categories. The main works are being evaluated on Caltech-101, a well-known database composed of 8677 images of objects of 101 different categories. Recognition accuracy achieved on this problem using 15 training images per category is between 50% and 60% (Grauman and Darrell 2007). However, all works based on the Caltech-101 data follow a traditional train and test approach, rather than focusing on interactive agents with on-line learning capabilities.

In this paper, we present a learning architecture that can be used by robotic agents for long-term and open-ended category acquisition. In this learning architecture, multiple object/category representations and multiple classifiers and classifier combinations are used. All learning computations are carried out during the normal execution of the agent, which allows continuous monitoring of the performance of the different classifiers. The measured classification successes of the base classifiers are used to dynamically reconfigure some of the classifier combinations as well as to select the classifier that will be used to predict the category of a new unseen object.

Agent Architecture

The developed agent is a computer with an attached camera running appropriate perceptual, learning and interaction procedures. The agent's world includes a user, a visually observable area and real-world objects whose names the user may wish to teach. The user, who is typically not visible to the agent, will therefore act as instructor. The user can change de content of the scene, by adding or removing objects.

Using a simple interface, the user can select (by mouse-clicking) any object from the visible scene, thereby enabling shared attention. Then, the user can perform the following teaching actions:

— Teach the object's category name
— Ask the category name of the object, which the agent will predict based on previously learned knowledge
— If the category predicted in the previous case is wrong, the user can send a correction.

Proc. LangRo'2007
978-972-96895-2-9

Main blocks

The agent architecture (figure 1) consists of a perception module, an internal lifelong category learning and recognition module and an action module. The current action system abilities are limited to reporting the classification results back to the user.

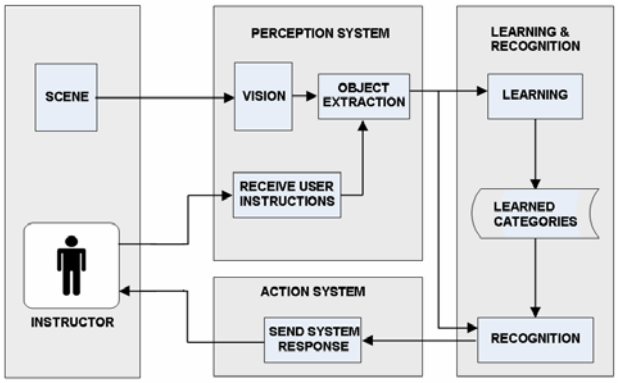


Figure 1 – Agent architecture

The tasks of the perception system include receiving user instructions, capturing images from the camera and extracting object features from images (figure 1). When the user points the mouse to an object in the scene image, an edge-based counterpart of the whole image is generated. The implementation of the canny algorithm, from the publicly available openCV library of vision routines, is used for edge detection. From this edges image, the boundary of the object is extracted taking into account the user pointed position. This is performed using a region growing algorithm and currently assumes that objects don't overlap (or occlude each other) in the image.

Given the boundary of the object, an edges-based counterpart of the object image is extracted from the full scene image (see example in Figure 2).

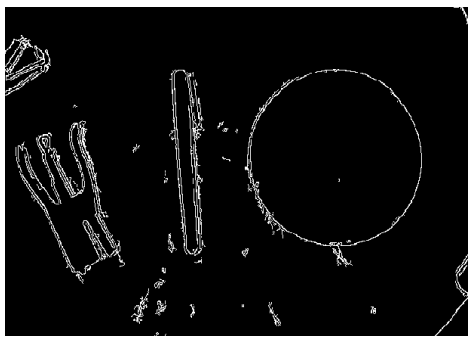


Figure 2 – Edges-based counterpart of a visual scene with three objects

Most of the features used by the classifiers described later in this paper are shape features extracted from this edges image. Only one classifier uses color features. In this case, the original object image is converted to the HSV color format. The primary purpose of using HSV format lies in the fact that most of the color information is in the H component (hue specifies the dominant wavelength of the color in most of its range of values), thus facilitating image analysis based on a single dimension.

The communication between the agent and the human instructor is supported by the perception and action systems. At present, the communication capabilities of the robot are limited to reading the teaching options (teach, ask, correct) in a menu-based interface and displaying classification results. In the future, simple spoken language communication will be supported.

Learning architecture

Language acquisition is highly dependent on the representations and methods used for category learning and recognition. Learning a human language will require the participation of the human user as teacher or mediator (Steels and Kaplan 2002; Seabra Lopes and Chauhan, 2007). A learning system in a robot should support long-term learning and adaptation. Such a system should support supervised, incremental, on-line, opportunistic and concurrent learning and should also be able to improve or optimize its performance through meta-learning (Seabra Lopes and Wang 2002; Seabra Lopes and Chauhan 2007).

The learning architecture proposed here (see Figure 2) was designed to satisfy these requirements. By organizing its categories and instances according to user's feedback, it behaves in a supervised way. It is on-line because it is integrated in the normal activity of the agent. It is incremental and opportunistic because it is able to adjust categories when new instances are observed, rather than requiring that training instances are given in a training phase or according to a pre-defined training schedule. It doesn't involve heavy computations, which facilitates the concurrent handling of multiple learning problems.

This learning architecture is based on the idea that using multiple representations, multiple classifiers and multiple classifier combinations, all potentially complementary of each other, can enhance global performance. Some of these ideas, particularly the use of classifier combinations, are not new in the machine learning literature (Xu et al. 1992). The main innovation in this architecture is that those complementarities are explored in an on-line learning architecture, and a simple form of meta-learning takes advantage of the on-line nature of the learning process, to improve global performance. Teaching and corrective feedback from the human mediator are used to monitor the classification success of the individual classifiers. The measured classification successes of the individual classifiers are used to dynamically reconfigure some of the classifier combinations as well as to select the classifier that will be used to predict the category of a new unseen object.

Feature Spaces

Objects should be described to the learning and classifications algorithms in terms of a small set of informative features. A small number of features will shorten the running time for the learning algorithm.

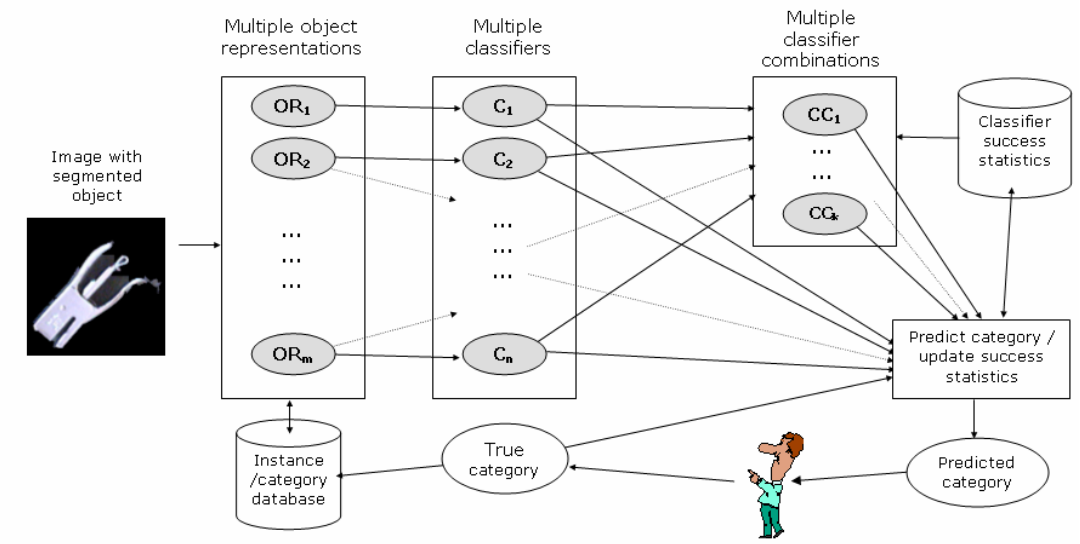


Figure 3 – Learning architecture

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

In the approach of this paper, multiple, possibly complementary feature spaces are concurrently explored. Most of these feature spaces result of segmenting the smallest circle enclosing the edges image of the object and centred in its geometric centre. For different feature spaces, such circle is segmented either into a number of slices (Figure 4, left) or a number of concentric layers (Figure 4, right). Current implementation uses 40 slices and 160 layers. Feature spaces based on this kind of segmentation are aimed at capturing shape information. In the following, feature spaces are briefly described.

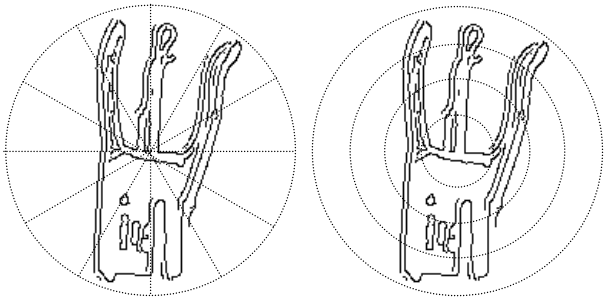


Figure 4 – Segmentation of edges image of an object into slices (left) and layers (right)

Shape slices histogram (SSH). The histogram contains, for each slice, the percentage of edge pixels in that slice with respect to the total number of edge pixels of the object. An example is given in figure 5a for the three objects shown in figure 2. Given the rotation-dependent nature of this feature space, similarity (or distance) between instances must be computed as the maximum similarity (or minimum distance) between the respective feature vectors as they are circularly rotated relative to each other.

Area (AREA). This feature space is composed of a single feature, *area*, defined as the total number of pixels of the object. This is the only scale-dependant feature space used in this work.

Shape slices normalized radii averages (SSNRA). For each slice, i, the average radius of all pixels in that slice, R_i, is computed. In this feature space, an object represented by a vector $\bar{r} = r_i \,..\, r_{40}$, where $r_i = R_i / R$ and R is the average of all R_i. This is the core of the feature space used in previous work (Seabra Lopes and Chauhan, 2007). An example is given in figure 5b for the three objects shown in figure 2. As in shape slices histogram, similarity computations involve rotations.

Normalized radius standard deviation (RADSD). This is another feature space composed of a single feature. Its value is the standard deviation of the normalized radii averages, $r_i \,..\, r_{40}$, mentioned in the previous paragraph.

Shape slices normalized radii standard deviations (SSNRSD). For each slice, i, the radius standard deviation of all pixels in that slice, S_i, is computed. In this feature space, an object is represented by a vector $\bar{s} = s_i \,..\, s_{40}$, where $s_i = S_i / R$ and R is the average radius as mentioned above. An example is given in figure 5c for the three objects shown in figure 2. As in other representations based on shape slices, similarity computations involve rotations.

Shape layers histogram (SLH). The histogram contains, for each layer, the percentage of edge pixels with respect to the total number of edge pixels of the object. This feature space is not only scale-invariant, but also rotation-invariant. An example is given in figure 5d for the three objects shown in figure 2.

Color ranges (COLOR). In this feature space, an object is represented by a set of the main colors of the object. Each color is represented as a range of hue values in HSV color space. These color ranges are

Proc. LangRo'2007
978-972-96895-2-9

extracted from a color histogram using a simple method presented in a previous paper (Seabra Lopes et al 2007).

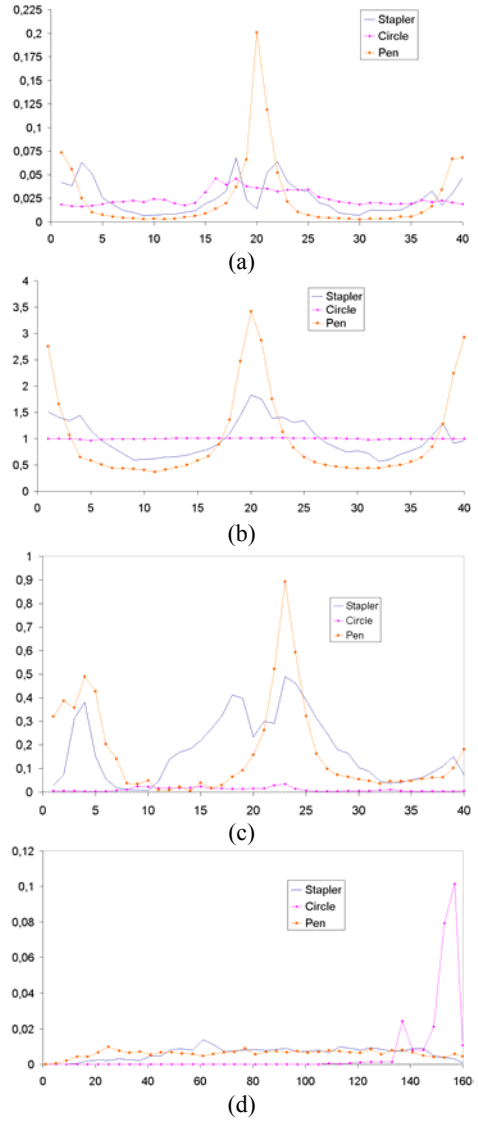


Figure 5 – Different types of shape features for the three objects in figure 2: (a) Shape slices histogram; (b) Shape slices normalized radii averages; (c) Shapes slices normalized radii standard deviations; (d) Shape layers histogram.

Similarity and Membership Measures

Categorization of a new previously unseen instance involves ranking the known categories according to measures of membership of the instance to each of the categories. In turn, computing membership measures often involves evaluating similarities and/or distances between instances.

Euclidean measures

Since objects are represented as feature vectors in most of the features spaces described above, an obvious similarity measure is inverse Euclidean distance. For two objects $\bar{x}$ and $\bar{y}$, with distance $D = \|\bar{x} - \bar{y}\|$, inverse Euclidean distance is given by $1/D$.

In instance-based classifiers, membership to a category is evaluated by computing and combining the similarities of the target object to the known instances of the category. In the present work, assuming that categories are homogeneous, i.e. that there are no significant intra-class variations, averaging the similarities of the target object to the known instances of the category seems an appropriate strategy for computing membership measures.

One of the membership measures used in this work is, therefore, computed by inverting and normalizing the average Euclidean distance of the target object to the instances of the given category C_i, as follows:

$$EuclidMem(C_i) = \frac{N}{D_i \sum_{k=1}^{N} (1/D_k)},$$

where N is the number of categories, i, k=1, …, N, and D_i and D_k are the average Euclidean distances of the target object to the known instances of categories C_i and C_k, respectively. The membership values $EuclidMem(C_i)$ sum to 1.0, allowing their use as evidence in Dempster-Shafer combinations.

Multi-resolution measures

In the present work, similarity is alternatively measured through a multi-resolution matching algorithm similar to the matching algorithm used in the recently proposed *pyramid match kernel* (Grauman and Darrell 2007). This kernel function was designed to enable the application of kernel-based learning methods to domains where objects are represented by unordered and variable-sized sets of features, such as sets of local features in computer vision. In this kernel, each feature set is mapped to a histogram pyramid, i.e. a multi-resolution histogram preserving the individual features distinctness at the base level. Then, the histogram pyramids are matched using a weighted histogram intersection computation.

The feature spaces used in the present work, as described above, are ordered and have a constant dimension, so mapping these representations to multi-resolution pyramids is direct. Then, the same basic matching algorithm can be applied.

The pyramid match score for two objects $\bar{x}$ and $\bar{y}$ is given by:

$$P_{\Delta}(\bar{x}, \bar{y}) = \sum_{i=0}^{L-1} w_i N_i(\bar{x}, \bar{y})$$

where L is the number of pyramid layers, $w_i = 2^i$ is the weight of layer i and N_i measures the additional matching at layer i, as given by:

$$N_i(\bar{x}, \bar{y}) = I(F_i(\bar{x}), F_i(\bar{y})) - I(F_{i-1}(\bar{x}), F_{i-1}(\bar{y}))$$

where $F_i(\bar{x})$ is the feature representation of object $\bar{x}$ at layer i and $I()$ is an intersection function which measures the overlap of two objects as follows:

$$I(A, B) = \sum_j \min(A_i, B_i)$$

Note that this type of matching applies not only to histograms, as done by Grauman and Darrell, but also to other feature vectors that are normalized by some constant R, as it happens in the "shape slices normalized radii averages" feature space described above. The use of pyramid matching in the present work extends a previous, simpler idea of the authors, which consisted of including in feature spaces block averages computed based on base-level ordered feature vectors (Seabra Lopes and Chauhan 2007; Seabra Lopes and Camarinha-Matos 1998).

Based on the pyramid match score, the following category membership measure for a particular target object and category C_i can be computed:

$$PyramidMem(C_i) = \frac{N \cdot P_i}{\sum_k P_k}$$

where N is the number of categories, i, k=1, …, N, and P_i and P_k are the average pyramid match scores of the target object to the known instances of categories C_i and C_k, respectively. The $PyramidMem(C_i)$ membership values sum to 1.0, allowing their use as evidence in Dempster-Shafer combinations.

Categorization

In the present work, categories are simply represented by sets of known instances. The known instances that are stored are those explicitly taught by the human user and also those objects that the agent failed to categorize correctly, leading to corrective feedback from the user. The agent, therefore, doesn't add to its instance database those objects that it was able to categorize correctly.

Categorizing a new previously unseen object involves computing measures of membership of the object to the known categories. The category with highest membership measure for the target object is returned. These computations are carried out by classifiers. In the present work, multiple classifiers and multiple classifier combinations are used.

Base classifiers

The use of a specific membership measure with a specific feature space results in a specific "base classifier". The following base classifiers were included in the implementation:

- Classifiers using single-dimension feature spaces with Euclidean membership measurement: "Area" (AREA); "Normalized radius standard deviation" (RADSD).
- Classifiers using feature vectors with Euclidean membership measurement: "shape slices histogram" (SSH-EM), "shape slices normalized radii averages" (SSNRA-EM), "shape slices normalized radii standard deviations" (SSNRSD-EM) and "shape layers histogram" (SLH-EM).
- Classifiers using feature vectors (the same as in the previous group) with pyramid membership measurement: SSH-PM, SSNRA-PM, SSNRSD-PM and SLH-PM.
- Classifier based on a color-based category representation and membership measure (COLOR) presented elsewhere (Seabra Lopes et al. 2007).

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

Classifier combinations and meta-learning

The complete learning and categorization approach includes classifier combinations. Some of the classifier combinations are dynamically reconfigured according to the observed success of the base classifiers. Classification success rates computed over the last N iterations (N=50 was used) is the performance measure used to guide reconfiguration of classifier combinations. This introduces a meta-learning component in the category learning system.

Dempster-Shafer combinations

The Dempster-Shafer theory of evidence is a powerful tool for representing and combining uncertain knowledge (Shafer 1976). It is based on a basic belief assignment, i.e. a mass function $m(A)$ that assigns a value in [0,1] to every subset A of a set of mutually exclusive propositions θ. The belief in the composite proposition $B \subseteq \theta$ is given by the sum of $m(A)$ for all $A \subseteq B$. The belief in θ sums to 1.0. In this theory, when multiple evidences allow to derive multiple basic belief assigments, these evidences can be combined. In particular, two basic belief assignments m_1 and m_2 can be combined by the following rule:

$$m(C) = \frac{\sum_{A,B,A \cap B=C} m_1(A) \cdot m_2(B)}{1 - \sum_{A,B,A \cap B=\varnothing} m_1(A) \cdot m_2(B)}$$

This rule is the basis of a well known method for combining multiple classifiers (Xu et al. 1992; Al-Ani and Deriche, 2002). Each classifier provides evidence that is expressed as a basic probability assignment. In the work of this paper, the membership measures described above (Euclidean-based and pyramid-based) are directly used as masses. As mentioned before, these membership measures are normalized to sum to 1.0.

Sets containing more than one category are assigned a mass of 0.0, so the approach comes close to the Bayesian combination approach. The main difference is that normalized membership measures are used instead of conditional probabilities. These conditional probabilities could be estimated based on the confusion matrixes of each classifier. The classical way of doing this is to acquire a confusion matrix for each classifier in a preliminary training/testing phase. This approach, however, is not viable in a long-term / open-ended learning scenario. Is such a scenario, therefore, the alternative would be to build the confusion matrixes on-line. This would imply that, in an initial stage as well as after the introduction of a new category, the conditional probabilities would be heavily biased by the specific cases seen so far. We did some exploratory experiments in this direction and observed that classifier combinations based on conditional probabilities start

behaving poorly, but eventually catch up with classifier combinations based on membership measures. However, even in the long run, conditional probabilities didn't seem to be able to significantly outperform membership measures, as far as classifier combinations are concerned.

Four Dempster-Shafer classifier combinations were included in the implementation, namely combinations of the top two, three, four and five most successful classifiers (respectively DS2TOP, DS3TOP, DS4TOP and DS5TOP). Since the classification success of each classifier is re-evaluated in each teaching/learning interaction with the human user, these classifier combinations are also dynamically reconfigured in each such opportunity.

Majority voting combinations

Voting methods are also well known in classifier combinations (Xu et al, 1992; Kittler et al, 1998). In the implementation, two dynamically reconfigured classifier combinations based on majority voting were included: majority voting of the top three and five most successful classifiers (respectively MAJ3TOP and MAJ5TOP). In addition, a classifier combination based on majority voting of all previously described classifiers (MAJORITY-ALL) was also included.

The Predicted Category

The internal computations described until now culminate in a category prediction that is communicated to the interlocutor(s) of the agent, typically a human user. This category will be the category predicted by the currently most successful classifier, considering all base classifiers and classifier combinations described above.

Experimental Evaluation Protocol

The word/category learning literature has some common features. One of them is the limitation on the number of learned words. The known approaches have been demonstrated to learn up to 12 words.

The other common feature is the fact that the number of words is pre-defined. This is contrary to the open-ended nature of the word learning domain. Then, given that the number of categories is pre-defined, the evaluation methodology usually consists of extracting certain measures on the learning process (Roy and Pentland 2002; Steels and Kaplan 2002; Yu 2005; Skocaj et al. 2007; Lovett et al, 2007). Some authors plot this type of measures versus training time or number of examples. As the number of words/categories is pre-defined, the plots usually show a gradual increase of these measures and the convergence to a "final" value that the authors consider acceptable.

However, robots and software agents are limited in their perceptual abilities and, therefore, cannot learn arbitrarily large numbers of categories, particularly when perception does not enable the detection of small between-category differences. As the number of categories grows, learning performance will evolve, with phases of performance degradation followed by recovery, but will eventually reach a breakpoint.

A well-defined teaching protocol can facilitate the comparison of different approaches as well as the assessment of future improvements. With that in mind, the teaching protocol of figure 6 was previously proposed (Seabra Lopes and Chauhan 2007). For clarity, its presentation is repeated here.

```
introduce Class0;
n = 1;
repeat {
   introduce Classn;
   k = 0;
   repeat {
      Evaluate and correct classifiers;
      k ← k + 1;
   } until ( ( average precision >
                 precision threshold and k≥n)
             or
            (user sees no improvement in precision));
   n ← n + 1;
} until (user sees no improvement in precision).
```

Figure 6 – Experimental evaluation protocol

This protocol is applicable for any open-ended class learning domain. For every new class the instructor introduces, the average precision of the whole system is calculated by performing classification on all classes for which data descriptions have already been learned. Average precision is calculated over the last $3\times n$ classification results (n being the number of classes that have already been introduced). The precision of a single classification is either 1 (correct class) or 0 (wrong class). When the number of classification results since the last time a new class was introduced, k, is greater or equal to n, but less than $3\times$n, the average of all results is used. The criterion that indicates that the system is ready to accept a new object class is based on the precision threshold.

Experimental Results

Experiments were conducted according to this protocol. The set of categories and the set of training instances were not established in advance. As categories were learned, new objects were fetched from the surrounding office environment and used to introduce new categories. Many objects were brought from the homes of the authors for proceeding with the experiments until the breakpoint was reached.

The experiments lasted for several days and a total of 3767 question/correction iterations (figure 7). In total, it was possible to teach 68 categories of real-world objects, which can be roughly grouped as follows: office objects – 40%; child toys – 20%; other home objects – 20%; other – 20%. Figure 8 displays one sample image per category. During the

teaching/learning process, the agent stored a total of 1168 training instances.

Figure 7 displays the evolution of classification precision versus number of question/correction iterations. As observed in previous work (Seabra Lopes and Chauhan 2007; Seabra Lopes et al. 2007), classification precision degrades after the introduction of each new category, then eventually recovers.

Towards the limit of the category discrimination abilities of the agent, learning starts to take longer. From figure 7, we see that most categories were learned in the first ~2000 iterations (exactly 60 categories), while in the remaining ~1800 iterations it was possible to learn only 8 additional categories. The breakpoint is also clearly visible in figure 9 which displays the evolution of the average number of training instances per category versus the increasing number of categories. After learning the first 30 categories, the system had stored less than 4 instances per category. While learning additional 30 categories, the number of instances per category continued to grow according to a linear trend, reaching an average close to 9 instances per category. Finally, while learning the last 8 categories, the number of instances per category abandoned the linear evolution trend and jumped to 17.

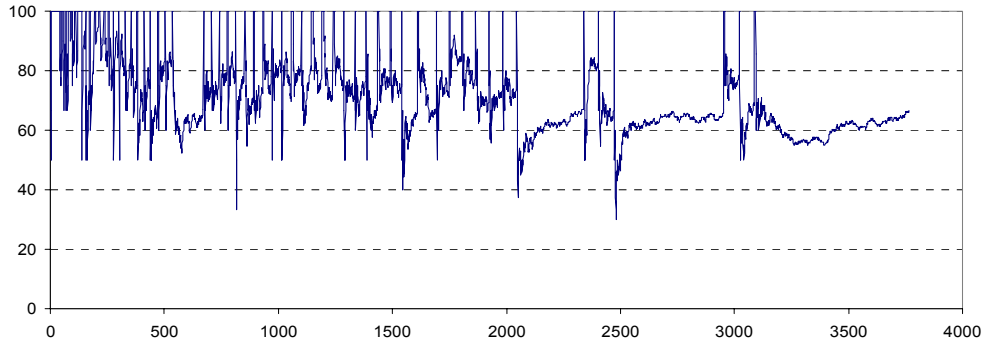


Figure 7 – Evolution of classification precision versus number of question/correction iterations

Figure 8 – Sample images of all acquired categories

Proc. LangRo'2007
978-972-96895-2-9

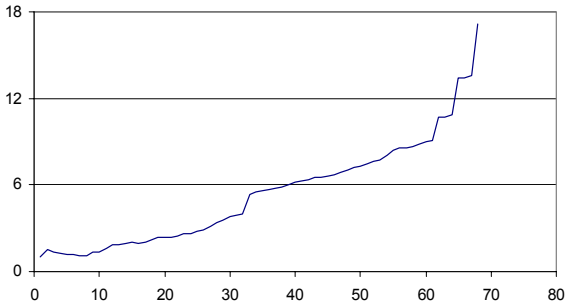


Figure 9 – Evolution of number of training instances per category versus number of acquired categories

Table I – Classification success rates for all classifiers

Single dimension classifiers	
AREA	44.6
RADSD	27.4
Shape feature vectors with Euclidean membership	
SSH-EM	2.8
SSNRA-EM	2.9
SSNRSD-EM	2.0
SLH-EM	30.8
Shape feature vectors with pyramid membership	
SSH-PM	46.2
SSNRA-PM	64.6
SSNRSD-PM	8.2
SLH-PM	43.1
Color-based classifier	
COLOR	8.7
Dempster-Shafer combinations	
DS2TOP	57.3
DS3TOP	57.5
DS4TOP	56.9
DS5TOP	64.2
Majority voting combinations	
MAJ3TOP	65.3
MAJ5TOP	63.9
Majority voting of all other classifiers	
MAJORY-ALL	70.6

The analysis of the performance of the individual classifiers is also relevant (Table I). Classifiers based on Euclidean membership perform very poorly. The best classifier in this group was SLH-EM (shape layers histogram with Euclidean membership) with an average precision of 30%. One of the single dimension classifiers performed better than that (AREA, 45%). Classifiers based on pyramid membership measurement performed far better than the Euclidean ones (e.g. SSNRA-PM, 65%).

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

The externally observable performance of the agent was very close to MAJORITY-ALL, exactly 70.0%. Note that, as mentioned before, the predictions of the agent are those of the current most successful classifier. In 64% of experiment time, MAJORITY-ALL was the best classifier. Other classifiers were the most successful for shorter amounts of time: MAJ3TOP with a share of 10%, SSNRA-PM with a share of 8% and DS5TOP with a share of 4%.

Conclusions

This paper presented a category learning architecture with several innovations. One of them is the use of multiple representations, multiple classifiers and multiple classifier combinations, all potentially complementary of each other. Although common in off-line learning, this approach has not been explored for on-line learning methods. Another innovation is the use of an attentional selection mechanism to reconfigure classifier combinations as well as to select the classifier that is used in a specific situation. Although our goal is not to emulate human category learning, some parallels can be drawn with research in that field (Ashby and Obrien 2005; Kruschke 2005). In particular, researchers have been recently moving toward the conclusion that human category learning relies on multiple memory systems and multiple representations. The also recently emphasized role of attentional selection, i.e. a mechanism of focusing on specific features or representations based on recent experience, can be paralleled with our mechanism of dynamically selecting and reconfiguring classifiers.

The shape-based feature spaces are also an original proposal of the authors in an attempt to develop computationally light classifiers that can be used in an on-line classifier combination architecture. The application of the pyramid matching algorithm of (Grauman and Darrell, 2007) to feature spaces where objects described, not by histograms, but by other normalized feature vectors is also a contribution of this paper, which actually produced excellent results. The extreme differences in performance between SSNRA EM (2.9%) and SSNRA-PM (65%) illustrate this point.

Overall, our approach seems to outperform several previous works initially cited. While previous approaches enabled learning of up to 12 categories, the proposed approach enabled learning of 68 categories in a long-duration experiment.

Besides the overall success of this work, compared to previous works with similar goals, the results provide support to some of the "ingredients" of the approach. In particular, the use of pyramid matching proved far more effective than Euclidean distance in similarity assessment. Also, majority voting proved successful in maximizing overall performance. The results don't provide irrefutable evidence in favor of the proposed attentional selection mechanism. Actually, selecting the "current best" classifier led the agent to perform slightly worse (70%) than the majority voting classifier (70.6%). Future experiments will be designed to enable drawing conclusive results concerning attentional selection.

Aknowledgements

The Portuguese Research Foundation (FCT) supported this work under contract POSI/SRI/48794/2002 (project "LANGG: Language Grounding for Human-Robot Communication"), which is partially funded by FEDER.

References

Al-Ani, A. and Deriche, M., (2002) A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence, *Journal of Artificial Intelligence Research*, 17, 333-361.

Ashby, F.G. and J.B. Obrien (2005) Category Learning and Multiple Memory Systems, *Trends in Cognitive Science*, 9(2), 83-89.

Barsalou, L. (1999). Perceptual symbol systems, *Behavioral and Brain Sciences*, 22(4), 577-609.

Cangelosi, A. & Harnad, S. (2000). The adaptive advantage of symbolic theft over sensorimotor toil: Grounding language in perceptual categories. *Evolution of Communication*, 4(1), 117-142.

Cowley, S. J. (2007). Distributed language: Biomechanics, functions and the origins of talk. In C. Lyon, C. Nehaniv & A. Cangelosi (Eds.), *Emergence of communication and language*. Springer, 105-127.

Fong, T., Nourbakhsh, I. & Dautenhahn, K. (2003). A survey of socially interactive robots: Concepts, design, and applications, *Robotics and Autonomous Systems*, 42, 143-166.

Gillette, J. Gleitman, H., Gleitman, L. & Lederer, A. (1999). Human simulations of vocabulary learning, *Cognition*, 73, 135-176.

Grauman, K. and T. Darrell (2007) The Pyramid Match Kernel: Efficient Learning with Sets of Features. *Journal of Machine Learning Research*, 8, 725-760.

Harnad, S. (1990). The symbol grounding problem, *Physica D*, 42, 335-346.

Kitler, J., R.P.W. Duin and J. Matas (1998) On Combining Classifiers, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(3), 226-239.

Kruschke, J. K. (2005). Category Learning. In: K. Lamberts and R. L. Goldstone (Eds.), *The Handbook of Cognition*, Chapter 7, 183-201.

Love, N. (2004). Cognition and the language myth, *Language Sciences*, 26, 525-544.

Lovett, A., M. Dehghani and K. Forbus (2007) Incremental Learning of Perceptual Categories for Open-Domain Sketch Recognition, *Proc. Int. J.Conf. Artificial Intellig. (IJCAI'07)*, p. 447-452.

Polikar, R., Udpa, L., Udpa, S. S. & Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *31*(4), 497-508.

Roy, D., and A. Pentland (2002). Learning words from sights and sounds: A computational model, *Cognitive Science*, 26, 113-146.

Seabra Lopes, L. and L.M. Camarinha-Matos (1998) Feature Transformation Strategies for a Robot Learning Problem, *Feature Extraction, Construction and Selection. A Data Mining Perspective*, H. Liu and H. Motoda (edrs.), Kluwer Academic Publishers.

Seabra Lopes, L. & Chauhan, A. (2006) One-Class Lifelong Learning Approach to Grounding, *Workshop on External Symbol Grounding. Book of Abstracts and Papers*, Plymouth, UK, 15-23.

Seabra Lopes, L. and A. Chauhan (2007) How many Words can my Robot learn? An Approach and Experiments with One-Class Learning, *Interaction Studies*, 8(1), 53-81.

Seabra Lopes, L., A. Chauhan and J. Silva (2007) Towards long-term visual learning of object categories in human-robot interaction, *Proc. 2nd Intelligent Workshop on Intelligent Robotics - 13th Portuguese Conference on Artificial Intelligence (EPIA2007)*, to appear.

Seabra Lopes, L. and J.H. Connell (2001) Semisentient Robots: Routes to Integrated Intelligence, *IEEE Intelligent Systems*, vol. 16(5), p. 10-14.

Seabra Lopes, L., Teixeira, A. J. S., Quinderé, M. & Rodrigues, M. (2005). From robust spoken language understanding to knowledge acquisition and management. *Proc. Interspeech 2005*, Lisbon, Portugal, 3469-3472.

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

Skocaj, D., G. Berginc, B. Ridge, A. Stimec and N. Hawes (2007) A System for Continuous Learning of Visual Concepts, *Proc. Int. Conf. on Computer Vision Systems (ICVS2007)*, Bielefeld, Germany.

Steels, L. (2001) Language games for autonomous robots, *IEEE Intelligent Systems*, 16(5), 16-22.

Steels, L. & Kaplan, F. (2002). AIBO's first words: The social learning of language and meaning, *Evolution of Communication*, 4(1), 3-32.

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

Thomaz, A. L., and C. Breazeal (2006) Transparency and Socially Guided Machine Learningm, *Proc. 5th International Conference on Developmental Learning (ICDL)*.

Thrun, S. (1996). *Explanation-based neural network learning: A lifelong learning approach*. Boston, MA: Kluwer.

Xu, L., Krzyzak, A., Suen, C.Y. (1992) Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Trans. Systems, Man and Cybernetics*, 22 (3), 418-435.

Yu, C. (2005). The emergence of links between lexical acquisition and object categorization: A computational study, *Connection Science*, 17(3-4), 381-397.

A constraint-based model of grounded compositional semantics

Van den Broeck, Wouter J. M.
Sony Computer Science Laboratory, 6, rue Amyot,
Paris, 75005, France
wouter@csl.sony.fr

Abstract

This paper outlines a constraint-based system that enables artificial agents to interpret and conceptualise rich meaning which involves different concept types and semantic functions. Such compositional meaning consists of a network of semantic building blocks that bundle a semantic function together with the necessary concept grounding and learning methods. The semantic blocks are implemented as constraints, and the compositional meaning is represented as a constraint network. The interpretation of such meaning corresponds to constraint satisfaction. The conceptualisation is realised as a goal-directed construction of the constraint network. The concept acquisition is fully integrated in the interpretation and conceptualisation processes.

Introduction

A system that enables agents to talk about the world can be decomposed in three sub-systems: the *sensorimotor system*, the *conceptual-intentional system* and the *language system* (Hauser et al., 2002). Figure 1 shows how these sub-systems interact.

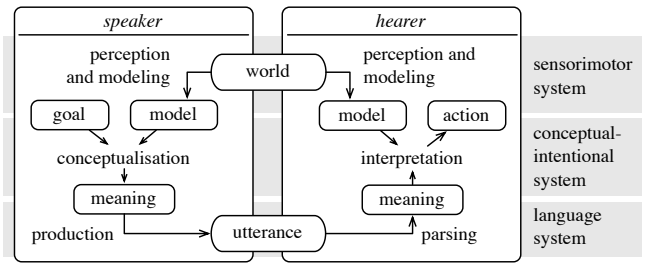


Figure 1: Overview of the interaction between the three sub-systems required for enabling artificial agents to communicate about the world through language.

The *sensorimotor system* takes care of the perception of the world and the construction of a model of that world. This world-model is used by the other sub-systems. The *language system* deals with the production of the utterance given the intended meaning, and the parsing of an utterance which yields the understood meaning. The third sub-system, the *conceptual-intentional system*, henceforth CIS, sits in between the language system and the sensorimotor system. It has to deal with the *representation*, *interpretation* and *conceptualisation* of meaning.

The *conceptualisation* process takes a speech-act goal and the world-model provided by the sensorimotor system. It composes the meaning that should be expressed in the utterance to be produced by the language system. The *interpretation* process takes the meaning reconstructed by the language system and interprets it in the context of the world-model provided by the sensorimotor system. The resulting action can range from executing the speaker's directive, or storing in memory the proposition in the speaker's assertive.

In this paper we focus on the conceptual-intentional system, and in particular on the question how such a system can be implemented for use in experiments involving language games (Steels, 1995).

To start we will consider the nature of the compositional meaning we want this system to be able to deal with. Such meaning consists of a network of semantic building blocks that take concepts as arguments. We will first look at the concepts and proceed with the semantic blocks.

Concepts

We will focus on speech-act goals that are concerned with the discrimination and/or description of objects in an observed scene. Figure 2 depicts a simple example scene which involves a number of objects with different shapes and sizes. If the speaker wants to draw the hearer's attention to object $o1$, then he/she could do so by saying "the pyramid". If the topic is rather object $o4$ then it could say "the big ball", while "the ball next to the big box" would do for object $o6$.

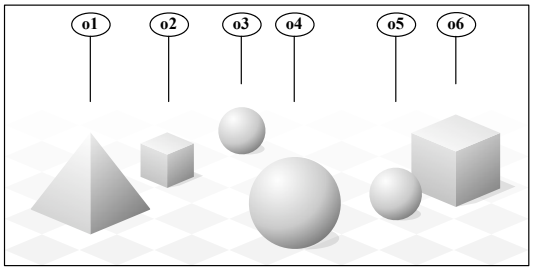


Figure 2: Simple example scene involving a number of objects, which are labeled for the purpose of the discussion.

Words like *ball*, *box*, *big*, *rightmost* and *next-to* each name a concept. Categories, prototypes, relations, events, roles, etc. are all different types of concepts.

Concepts can be used to discriminate specific objects by filtering them from a given context. A concept such as the shape prototype BALL can for instance be used to filter the objects that are ball-like, while a concept such as the comparison BIG can be used to filter the objects with a size larger than the average size.

Concepts that by themselves do not discriminate a topic can be combined. The phrase "the big ball" for instance, properly discriminates object *o*4 in the above example scene, even though there is more than one ball and several big things. There is however only one object that is both big and ball-like.

Concept grounding

The concepts need to be grounded in the sensorimotor functionality which interacts with the world. Different methods can be used for the grounding of concepts, for example neural networks are used in (Plunkett et al., 1992), probability density estimation in (Roy and Pentland, 2002), radial basis function networks in (Steels and Belpaeme, 2005), nearest neighbor (Belpaeme and Bleys, 2005), discrimination trees (Steels, 1996), event feature detectors (Siskind, 2001), etc.

A grounding method is minimally capable of assessing if some entity in the world-model belongs to some category. Each category for a particular grounding method corresponds to some particular set of *parameters*. Figure 3 lists some basic grounding methods and the kinds of parameters associated with the concepts grounded by the respective techniques.

Since no single grounding method is well suited for all types of concepts, the system needs to accommodate different grounding techniques.

Semantic functions

Concepts serve as arguments for *semantic functions* such as the context filtering discussed before. Other examples of semantic functions are: quantification as in "*the* ball" or "*some* boxes", set operations as in "the balls *and* the boxes" or "all balls *except* the rightmost", predication as in "the ball *is big*", negation as in "the box is *not* round", deictic reference as in "... *that* is round", etc. Note that different semantic functions can use the same concepts.

The artificial agents need to be able to autonomously interpret meaning that involves such semantic functions. Each semantic function thus requires a procedural implementation. This implementation takes the relevant concepts as arguments and calls the relevant grounding methods where needed.

Consider for example a semantic function that filters a set of entities according to a concept type that is grounded by means of a multi-layered perceptron. The application of this filtering involves the categorisation of each entity in the context by means of the perceptron, which is configured with the parameters – the weights – associated with the given concept. The results of these categorisations are then used by the semantic function to derive the filtered target-set[1].

Concept acquisition

Each agent has its own collection of concepts. These repertoires are furthermore not fixed. Agents need to be able to invent or learn new concepts or adapt existing ones. The nature of the learning methods depends on the type of the concerned concepts. Figure 3 lists a number of concept grounding methods and potential learning methods. Back propagation can for example be used with the multi-layered perceptron based grounding method.

grounding method	concept parameters	learning methods
k-NN	points, k	new point or shift points
ML perceptron	weights, θ	back-propagation
discrimination tree	segment	segmentation
...	...	...

Figure 3: This table lists some basic grounding methods and the corresponding concept parameters and learning methods. For the k-nearest-neighbours (k-NN) method, the parameters are one or more prototypical points in the data-space, and the (optional) value *k*. The learning method is either simply adding the positive example point or shifting the points based on positive and negative examples. For the multi-layered (ML) perceptron method, the parameters are the weights and (optionally) the threshold function, while the learning method is back-propagation. The third grounding method involves a discrimination tree (Steels et al., 2000), for which a concept corresponds to some segment (a node in the tree) or a set of segments.

A typical learning situation occurs when the speaker's utterance involves a word that the hearer does not know. Consider for instance the situation in which the speaker says "the frouple" to discriminate object *o*1 (the pyramid) in figure 2. The hearer does not know this word and indicates that it could not understand the utterance. The speaker could then draw the attention to the topic through other means, such as by pointing to it. This presents a learning opportunity for the hearer. It now knows the context and the topic, and could try to infer the concept that corresponds with the word "frouple". The candidate concepts are those that properly discriminate that topic. All candidates, or one chosen according to some heuristics such as the saliency, can then be passed to the learning method associated with the grounding method.

This inference of candidate concepts can be seen as a different operational mode of the involved semantic function. Where interpretation corresponds to taking a context

[1] or multiple candidate target-sets

Proc. LangRo'2007
978-972-96895-2-9

source-set and a concept to produce a filtered target-set, here the semantic function takes a source-set and a filtered target-set, and infers the concepts that could account for the filtering of that target-set from the source-set.

Compositional meaning

Rich meaning involves different types of concepts and a variety of semantic functions that take these concepts as arguments. The conceptual-intentional system has to be able to both interpret and conceptualise such compositional meaning. The building blocks of these compositions each bundle a semantic function together with the necessary grounding and learning methods, as shown in figure 4. The involved functionality is wrapped in a uniform, abstract interface. This abstraction enables the semantic *composer* to transparently combine disparate underlying functionality.

The interface of a semantic block provides one slot for each argument that the involved semantic function needs to operate over. Compositional meaning is constructed by linking together the slots of multiple semantic blocks.

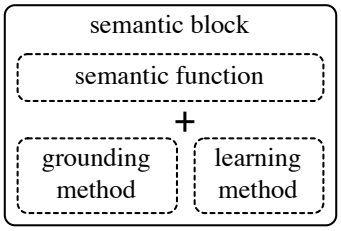


Figure 4: A semantic block combines a semantic function together with the necessary concept grounding and learning methods.

The tight coupling between the semantic functions and the grounding and learning methods affords a strong interaction between language use and concept formation. Such interaction is required to enable the structurally coupled evolution of language and concept repertoires.

Composition strategies

As mentioned earlier, the phrase "the big ball" properly discriminates object $o4$ in figure 2 because there is only one object that is both big and ball-like. The interpretation of this composition can be implemented by filtering in parallel the set of balls and the set of big things, and then taking the intersection of both sets.

This composition strategy is however not sufficient. Consider for instance the phrase "the big box" in the context of the scene shown below in figure 5. It discriminates object $o2$ even though the intersection of the set of big things $\{o1, o3\}$, and the set of box-like things $\{o2, o4\}$, is empty. Interpreting such phrase rather consists of first interpreting the noun relative to the context of the whole phrase. This yields a sub-context that consists of all boxes, i.e. $\{o2, o4\}$. Then the modifier is interpreted relative to this sub-context, which yields the bigger of both boxes, the intended topic.

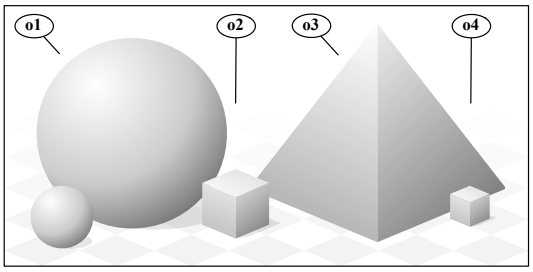


Figure 5: A scene with two big and three small objects.

The kind of context manipulation involved in the interpretation of a modifier-head structure can be made possible by providing (explicit) slots for the source-set (the input context) and the target-set (the filtered context) in the concerned semantic blocks. The modifier-head structure can then be attained by linking the target-set of the head's block to the source-set of the modifier's block. A block-diagram that represents this set-up is shown in figure 6.

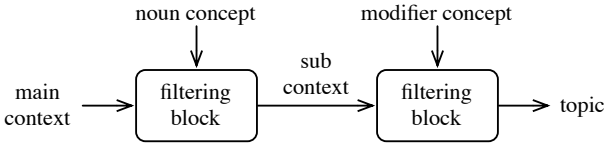


Figure 6: A semantic block-diagram for a noun + modifier phrase like "big box". The noun filtering block takes the main context and the noun concept and produces the filtered sub-context. This sub-context and the modifier concept are then used by the modifier building block to filter the topic.

Constraint networks

A semantic block generally supports several ways in which data flows in and out of that block. The concrete flow depends on the availability of values for the concerned slots. This availability is for instance different for a regular interpretation situation than for a learning situation. The ability to deal with different data-flows can be captured by implementing the semantic blocks as computational entities called *constraints*.

If the semantic blocks are implemented as constraints, then the compositional meanings correspond to constraint networks. Interpreting such meaning then corresponds to finding a solution for the constraint network, i.e. solving the *constraint satisfaction problem*.

A constraint can be represented as an n-slot predicate in which each slot is occupied by a variable. Multiple constraints form a network if slots from different constraints are occupied by the same variable.

Examples

Let's consider some examples of the interpretation and concept learning processes. These examples involve four types of semantic constraints, which are here represented

as n-slot predicates in which each slot is occupied by a variable. Multiple constraints form a network if slots from different constraints are occupied by the same variable.

The first two semantic constraints are *filter-set-prototype(target-set, source-set, prototype)* and *filter-set-size(target-set, source-set, comparison)*. They involve a filtering function such as described before. The first can filter the source-set by some prototype such as BALL or BOX. The second takes a comparison such as BIGGER-THAN, and retains in the target-set those objects from the source-set which are bigger than average.

The third block is *unique-element(object, set)*. This block asserts that the filler of the *set* slot is a set that contains one element; the filler of the *object* slot. It is used to cover the uniqueness of the topic. The fourth semantic block is *equal-to-context(set)*, which simply asserts that the filler of the *set* slot equals the set of objects in the observed context.

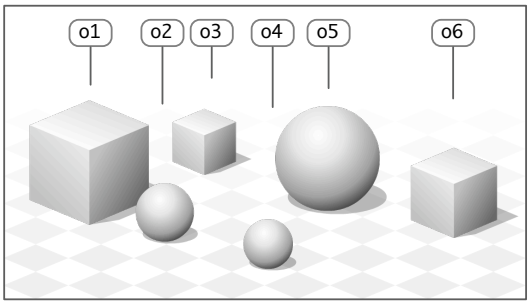


Figure 7: A scene with a number of object of varying size and shape.

Example 1: the big ball

Let's consider the semantic composition that discriminates object *o5* in the scene shown in figure 7. Combining these four constraints in a suitable composite meaning, gives the constraint network shown in figure 8.

Figure 8: Semantic composition example 1

```
{ equal-to-context(context),
  filter-set-prototype(set-1, context, prototype),
  filter-set-size(set-2, set-1, comparison),
  unique-element(topic, set-2) }
```

Interpretation Let's assume that the grammatical analysis of an utterance such as "the big ball" yields this composition plus the bindings: *prototype* ← BALL and *comparison* ← BIG, which are returned by the lexical look-up of "ball" and "big" respectively.

The semantic composition can now be interpreted by solving the constraint satisfaction problem. First the *equal-to-context* constraint binds the *context* variable to the complete set of objects in the scene, i.e. *context* ← $\{o1, o2, o3, o4, o5, o6\}$. Given the bindings for both the *context* and *prototype* variables, the *filter-set-prototype* constraint can infer a binding for *set-1*, i.e. the set of ball-like objects: $\{o2, o4, o5\}$. With this binding and the comparison, the *filter-set-average* constraint can now infer the binding *set-2* ← $\{o5\}$, since *o*5 is larger than the average size of the three balls. Finally, *unique-element* can correctly bind *topic* to *o*5, as such yielding the intended topic.

Acquisition Say we hear "the froople ball" but do not know the meaning of "froople". If we signal our misunderstanding to the speaker, and the speaker manages to draw our attention to the intended topic through other means, such as pointing, an opportunity for learning presents itself. We take the same semantic composition and fill in the known bindings: *prototype* ← BALL and *topic* ← *o*5. We can now again try to find a solution for the constraint network.

Applying the *unique-element* constraint gives the binding *set-2* ← $\{o5\}$. Applying the *equal-to-context* and *filter-set-prototype* constraints gives *set-1* ← $\{o2, o4, o5\}$. Given these bindings the *filter-set-size* block can try to abduct a comparison that could account for the filtering from the *set-1* to *set-2*. If this concept already exists in the inventory, a new entry between this concept and the form "froople" can be added in the lexicon. If it was not conceptualised before, it can also be added in the conceptual inventory.

Example 2: the ball next to the big box

As a second example we will assume the same context, but take *o*2 as the topic. We cannot easily find a semantic program that discriminates this topic using the same constraints as before. Let's therefore introduce an additional semantic block: *filter-set-relation(target-set, source-set, relation, referent)*. This block filters all elements from the source-set for which the relation does not apply with respect to the referent. The relations we consider here are spatial relations, such as NEXT-TO, or IN-FRONT-OF. This enables us to construct the semantic composition that corresponds to "the ball next to the big box", which properly discriminates the intended topic. The resulting composition is shown in figure 9.

Figure 9: Semantic composition example 2

```
{ equal-to-context(context),
  filter-set-prototype(set-1, context, proto-1),
  filter-set-size(set-2, set-1, comparison),
  unique-element(referent, set-2),
  filter-set-prototype(set-3, context, proto-2),
  filter-set-relation(set-4, set-3, relation, referent),
  unique-element(topic, set-4) }
```

For a regular interpretation the bindings are: *proto-1* ← BOX, *comparison* ← BIG, *proto-1* ← BALL, and *relation* ← NEXT-TO. Resolving the constraint network will first bind *referent* to *o*1 like in the previous example, and *set-3* to the set of balls, i.e. $\{o2, o4, o5\}$. Given these bindings

Proc. LangRo'2007
978-972-96895-2-9

the *filter-set-relation* block can now select from *set-3* those elements which are 'next-to' the referent and bind this set, i.e. $\{o2\}$, to *set-4*, giving us the correct topic.

Goal-directed composition of constraint networks

The conceptualisation of a semantic composition corresponds to the construction of a constraint network. The input for this process is a communicative goal, e.g. 'discriminate topic *X* in the sensory context', and an inventory of primitive constraints. The resulting constraint network has to be coherent and fulfil the given goal when interpreted by the hearer. In order for the hearer to be able to properly interpret the decoded composition, all arguments that cannot be inferred should be expressed in the utterance. These *essential* arguments thus have to be representable in language, for instance as lexical forms.

Finding a suitable constraint network given some goal is a combinatorial problem. Blindly trying to link together various constraints in arbitrary configurations and checking if the results satisfy the requirements is not a viable strategy. We propose a structured, goal-directed strategy to manage the combinatorial explosion.

For a semantic composition to be useable, it must be resolvable given the essential arguments. All other bindings in the solution must be directly or indirectly inferable from this select set of bindings. In other words, there must exist a directed, non-cyclic dependency network among the bindings which reflects the inferential flow from the essential source bindings to the binding or bindings that represent or otherwise contribute to the communicative goal. The process of creating an appropriate semantic composition can be guided by this requirement.

Let's for example consider the construction of the semantic composition shown in figure 8. The initial goal is to discriminate object $o5$ from the sensory context shown in figure 7. We start the composition by introducing a variable and bind the topic to it. This binding is meant to be inferable during interpretation, so we need to add a constraint that can infer the binding. Most constraints however hold over more than one variable, which will need to be added. The bindings for these new variables also need to be either essential bindings or be inferable themselves. Introducing a new constraint to fulfil a goal might thus introduce new sub-goals, which need to be fulfilled recursively.

Let's say we add *unique-element(topic, set-2)* to infer the topic. This introduces a new sub-goal: find support for (the binding of) *set-2*. Adding *filter-set-average(set-2, set-1, comparison)* fulfils this sub-goal, but yields two new sub-goals: *set-1* and *comparison*. The comparison concept can be expressed in the utterance, but the set will have to be recursively dealt with.

A complete overview of the composition process is shown in figure 10. Each row represents a step in the process, starting with the initial step in the first row. The first column gives the goal for each step. The second column shows the 'action' taken to fulfil the goal, which is either a new constraint or an argument that has to be expressed in the utterance. The third column lists the sub-goals entailed by adding a constraint. Each of these sub-goals needs to be fulfilled in one of the subsequent rows.

Figure 10: Goal directed composition

goal	constraint or argument	subgoals
topic	unique-element(topic, set-2)	set-2
set-2	filter-set-average(set-2, set-1, comparison)	set-1, comparison
comparison	BIG	-
set-1	filter-set-prototype(set-1, context, prototype)	context, prototype
prototype	BALL	-
context	equal-to-context(context)	-

The composition process starts with the initial goal and ends when all the sub-goals that were introduced along the way, are fulfilled. For each goal there might be several constraints that could infer that goal. The composition shown in figure 10 thus represents but one particular path of potentially many. All these paths form a tree. Various strategies can be used to more efficiently explore this tree. We for instance apply an eager search strategy based on a heuristic that favours smaller compositions, with less unfulfilled goals and a smaller amount of essential arguments. We prune branches that involve a cyclic dependency and try to prune inconsistent branches as soon as possible by propagating the constraints where possible after each extension.

Finally we would like to note that this composition mechanism can also deal with situations in which the structure of the semantic composition was not fully understood. It can be used to hypothesise on a plausible completion of an incomplete network by adding constraints to account for bindings not yet accounted for in exactly the same way as outlined before.

Conclusion

A semantic building block bundles all cognitive functionality that concerns a particular concept type. This includes both the concept formation functionality and the semantic operations that for instance categorise a set of visual stimuli. By encapsulating the procedural details and providing a uniform, abstract interface, different concept grounding techniques can be transparently combined.

Semantic blocks establish an omni-directional relationship between a number of arguments, which can be naturally implemented as constraints. A semantic composition can then be represented as a constraint network. The declarative nature of such constraint networks permits a flexible control-flow. This affords a natural and uniform treatment of various compositional production, interpretation and learning needs, as was shown in the examples.

The grounding of both the concepts and the basic semantic operations is taken care of by the semantic blocks. The semantic compositions attain their grounding from

their components and the procedurally embodied constraint satisfaction framework.

In sum, the proposed model satisfies the requirements outlined in the introduction. A fully operational implementation of this model has been developed and can be demoed upon request.

Acknowledgements

The research presented in this paper builds on ideas first introduced in (Steels, 2000) and elaborated on in (Steels and Bleys, 2005).

This research is supported by Sony Computer Science Laboratory in Paris and the ECAGENTS project funded by the Future and Emerging Technologies programme (IST-FET) of the European Community under EU R&D contract IST-2003-1940. The information provided is the author's sole responsibility and doesn't reflect the Commission's opinions. The Commission is not responsible for any use that may be made of data appearing in this article.

References

Belpaeme, T. and Bleys, J. (2005). Explaining universal color categories through a constrained acquisition process. *Adaptive Behavior*, 13(4):293–310.

Hauser, M. D., Chomsky, N., and Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298:1569–1579.

Plunkett, K., Sinha, C., Moller, M. F., and Strandsby, O. (1992). Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net. *Connection Science*, 4:293—312.

Roy, D. K. and Pentland, A. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science*, 26:113—146.

Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research (JAIR)*, 15:31–90.

Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial Life*, 2(3):319–332. (eds) Bedau, M.A. and Taylor, C.E. Cambridge, MA: The MIT Press.

Steels, L. (1996). Perceptually grounded meaning creation. In Tokoro, M., editor, *ICMAS96*. AAAI Press.

Steels, L. (2000). The emergence of grammar in communicating autonomous robotic agents. In Horn, W., editor, *ECAI2000*, pages 764–769, Amsterdam. IOS Press.

Steels, L. and Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28(4):469–89.

Steels, L. and Bleys, J. (2005). Planning what to say: Second order semantics for fluid construction grammars. In Bugarin Diz, A. and Reyes, J. S., editors, *Proceedings of CAEPIA '05. Lecture Notes in AI.*, Berlin. Springer Verlag.

Steels, L., Kaplan, F., McIntyre, A., and van Looveren, J. (2000). Crucial factors in the origins of word-meaning. In Dessalles, J.-L. and Ghadakpour, L., editors, *Proceedings of The 3rd Evolution of Language Conference*, pages 214–217, Paris. ENST 2000 S 002.

Speech Perception as Non-symbolic Pattern Recognition

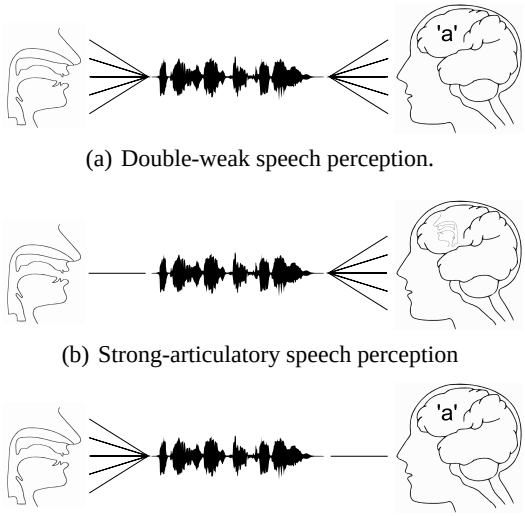
S. F. Worgan and R. I. Damper
Information: Signals, Images, Systems (ISIS) Research Group,
School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
sw205r@ecs.soton.ac.uk

Abstract

Despite ongoing research, the human ability of speech perception remains a mystery. Current phonetic theory is divided by two points of contention: the relationship from production to signal to audition and the object of perception/cognition. Here we discuss the role of current phonetic theory within this debate and propose our own hypothesis. We argue that human speech is enabled through loosely constrained articulation and audition coupled with the cognitive process of direct realism (DR). We also contend that disembodied pattern recognition is sufficient for the perception of phonetic tokens, as grounding can be maintained through the properties of real speech. However, to maintain this at the semantic level we feel that robotic embodiment will be necessary.

Although related to motor theory (MT), DR differs in a number of important ways. Significantly, speech perception is not held to be 'special' … "and there is no more reason to propose a role for the speech motor system in speech perception than to propose an analogous role for the viewer's locomotor system in the visual perception of walking" (Fowler, 1996, p. 1731). Instead of forming cognitive representations of the external world (either gestural or acoustic), our senses cause the direct perception of the gesture through the acoustic signal.

DR faces various criticisms, arising through its association with MT, as they are often treated as one and the same, e.g., Sussman (1989); Ohala (1996). Other criticisms are more specific. What is the force enabling auditory distinctiveness if we only perceive the gesture? Surely we would be driven to maintain *articulatory* distinctiveness? Fowler argues that the acoustic signal still conveys information about the gesture, which accordingly must be sufficiently distinct. But it does not follow that a distinct signal is evidence for a symbolic auditory representation. Another objection is that those who can't speak can still perceive speech. Motor theorists believe that an "innate vocal-tract synthesizer" (Liberman and Mattingly, 1985) can overcome this objection. While Fowler reemphasises that the direct perception of speech derives from a general theory of perception, this "inability to reproduce heard gestures does not imply that they did not perceive gestures (any more that the typical person's inability to perform a triple axel implies that he or she cannot see them)" (p. 1738). DR does not have to imply a motor theory of speech perception. It only needs to agree with MT in the trivial sense—we obviously 'perceive' the vocal tract as it is the source of the speech signal. Where DR can provide insight is in determining the object of speech perception.



(a) Double-weak speech perception.

(b) Strong-articulatory speech perception

(c) Strong-auditory speech perception

Figure 1: Conflicting phonetic theories use evidence of strong constraints on articulation or audition to argue for different symbolic systems of perception.

Using Nearey's (1997) framework, we can classify conflicting theories of perception into strong-auditory, strong-articulatory, double-strong and double-weak (see Figure 1). Strong-auditory theories include Stevens's (2002) well-known quantal theory. By contrast, strong-articulatory theories include MT and Fowler's direct realism. Double-weak theory defines a middle course, loosening constraints on both production and perception. However, many would consider it to be an auditory rather than articulatory theory.

Such disagreements arise because Nearey's classification only considers the means of production, the signal and perception of speech, whereas the current major source of disagreement is the form of the cognitive tokens. Auditory theories hold that these smallest tokens are resolved as

(a) Fowler's direct realism

(b) Proposed double-weak direct realism

Figure 2: A comparison of Fowler's direct realism and double-weak direct realism. The phonetic evidence suggests a double-weak approach, while our own work proposes a direct realist cognitive theory.

idealised symbolic phonetic tokens, whereas MT holds that the ultimate forms of perception are gestural tokens. Considered in these terms we can see that DR and MT (lumped together in Nearey's framework) are clearly different, as DR considers the perception of speech to be direct "unmediated by processes of hypothesis testing or inference making and unmediated by mental representations" (Fowler, 1996, p. 1731)—articulatory or acoustic. Freed from the need to lump all gesturalist theories into the strong-articulatory camp, we can see that DR is in fact a double-strong gesturalist theory (as opposed to motor theories strong-articulatory gesturalist approach). As clearly stated by Fowler: "phonological gestures are the public actions of the vocal tract that cause structure in acoustic speech signals. By hypothesis, they will be found to cause specifiers or invariants in the acoustic signal" (p. 1731).

We believe that speech is directly perceived; what is perceived (in the trivial sense) is the vocal tract. Although this appears to agree with Fowler, our theory differs in important respects. We question Fowler's naïve realism assertion that invariant "specifying acoustic properties is what allows perception of the phonological properties to be direct" (p. 1731). We feel that this plays into the hands of a number of arguments against the philosophy of DR. Rather we, like Nearey, are "genuinely impressed by the quality of the research by both auditorists and the gesturalists that is critical of the other position" (p. 3242). Given this we take a double-weak standpoint to the production and auditory perception of the speech signal. However, we do not believe that this double-weak approach necessarily precludes DR. As Figure 2(b) shows, in this new framework we can conceive of loosely-constrained articulation and perception coupled with the direct perception of speech, leading to a new double-weak direct realism. Clearly, there needs to be a de-coupling between the constraints on speech and the cognitive objects of perception.

To support this assertion, we have constructed a computational model that is able to acquire the phonetic structure of real speech using the details of this hypothesis. An artificial agent, equipped with a biologically plausible auditory system and vocal tract, is able to reproduce a range of phonemes after being exposed to real speech. Both its auditory and articulatory functions are loosely constrained (in accordance with double-weak theory) and at no time does it establish symbolic phonetic tokens with its cognitive abilities. Rather, complex auditory cues are used to enable the agent to reproduce the perceived phonemes. We can infer from this reproduction that the agent is capable of the direct perception of speech through pattern recognition. Why has this separation between the constraints present within the articulatory gesture and auditory signal not taken place before? Perhaps because evidence for a highly constrained vocal tract has been assumed to be evidence for abstract gestures as the objects of perception. Accordingly, a highly-constrained acoustic signal has been assumed to be evidence for abstract phonetic tokens. We argue that this is not necessarily the case.

Direct realism supposes that speech is perceived directly, in the absence of any idealised abstract tokens—either phonetic or articulatory. To test this hypothesis, our agents have been embodied in a real-speech environment avoiding the current symbolic phonetic systems which force a (potentially-ungrounded) symbolic solution. To develop our theory from the phonetic to the syntactic level, and to avoid a reversion to ungrounded symbolism, we will need to ground the evolved phonemes in real speech and the evolved syntax in the real world. Thus, future work will develop robotic agents to test further our notions of DR within language. Ultimately, DR has lead us to believe that the continued modelling of language will require embodiment through the use of robotics.

References

Fowler, C. A. (1996). Listeners do hear sounds, not tongues. *Journal of the Acoustical Society of America*, 99(3):1730–1741.

Liberman, A. M. and Mattingly, I. G. (1985). The motor theory of speech perception revised. *Cognition*, 21(1):1–36.

Nearey, T. M. (1997). Speech perception as pattern recognition. *Journal of the Acoustical Society of America*, 101(6):3241–3254.

Ohala, J. (1996). Speech perception is perceiving sounds not tongues. *Journal of the Acoustical Society of America*, 99(3):1718–1725.

Stevens, K. N. (2002). Toward a model for lexical access based on acoustic landmarks and distinctive features. *Journal of the Acoustical Society of America*, 111(4):1872–1891.

Sussman, H. (1989). Neural coding of relation invariance in speech: Human language analogs to the barn owl. *Psychological Review*, 96(4):631–642.

Towards Generating Referring Expressions in a Mobile Robot Scenario

Hendrik Zender and Geert-Jan M. Kruijff*
Language Technology Lab, German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken Germany
{zender,gj}@dfki.de

Abstract

This paper describes an approach towards generating referring expressions that identify and distinguish spatial entities in large-scale space, e.g. in an office environment, for autonomous mobile robots. In such a scenario a dialogue is often about things and places outside the current perceptual fields of the interlocutors. One of the challenges therefore lies in determining an appropriate dialogue context. Other important issues are to have adequate models of both the large-scale spatial environment and of the user's knowledge.

Introduction

In earlier work, we have presented a conversational autonomous mobile robot (Zender et al., 2007), emphasizing situated dialogue for teaching the robot about its environment. Besides understanding human-like concepts the robot must be able to express itself in a way that is understandable by humans. It is therefore crucial that the robot can produce expressions that successfully refer to entities in its environment.

Previous approaches to the generation of referring expressions (GRE) in the general domain of conversational agents have mainly focused on small-scale scenes or closed-context applications, (Kelleher and Kruijff, 2006), (Funakoshi et al., 2004), (Horacek, 1997), (Dale and Reiter, 1995). Although there are well-established methods for generation referring expressions from both explicit and implicit scene models, only limited research has focused on how to determine what part of a scene constitutes the current context. This is of special importance when conducting a situated dialogue about *large-scale space*, where large-scale space is defined as "a space which cannot be perceived at once" (Kuipers, 1977). For the dialogue this means that most potential referents and distractors are not in the visual fields of the interlocutors, but still they will want to talk about them.

In this paper, we present an approach to adapt the *incremental algorithm* (Dale and Reiter, 1995) to a scenario where a conversational robot has to refer to spatial entities in large-scale space. We will show how our approach of Conceptual Spatial Mapping (Zender and Kruijff, 2007) both provides a suitable knowledge base for the algorithm and serves as a basis for determining the context set.

* The research reported of in this paper was supported by the EU FP6 IST Cognitive Systems Integrated project *Cognitive Systems for Cognitive Assistants* "CoSy" FP6-004250-IP.

Background

The task of generating referring expressions can be paraphrased as finding a description for an entity in the world (the *intended referent*) that refers to the intended referent and only the intended referent. This implies that the description must be chosen in a way that prevents it from referring to another entity in the current *context set*. All entities in the context set except the intended referent form the *contrast set*. The referring expression must thus distinguish the intended referent from the members of the contrast set. A referring expression is a noun phrase (NP) of any degree of complexity. In order to provide enough information to uniquely identify the intended referent, further attributes of the referent need to be expressed, for instance with adjectives or prepositional phrases, which in turn might contain a referring expression NP.

The *incremental algorithm* of (Dale and Reiter, 1995) constitutes an approach to the GRE problem, which they rephrase in terms of the *Gricean Maxims*. Inherently, any referring expression should fulfill the Maxim of Quality in that it should not contain any false statements. The algorithm also ensures that only properties of the referent that have some discriminatory power are realized (Maxim of Relevance). Moreover, they try to fulfill the Maxims of Manner and Quantity in that the produced expressions are short and do not contain redundant information. The incremental algorithm provides a solution to the GRE problem with a reasonable run-time complexity. This is achieved by not trying to find an optimal referring expression, which Dale and Reiter justify by findings in psycholinguistics.

Since we are going to present our approach in terms of the incremental GRE algorithm, it is important to briefly explain its relevant principles (cf. Algorithms 1, 2, 3). The algorithm needs a knowledge base that describes the *properties* of the domain entities through *attributes* and *values*. A special attribute is an entity's *type*. The algorithm is initialized with the *intended referent*, a *contrast set* (defined as the *context set* without the intended referent) and a list of *preferred attributes*. It then iterates through this attribute list in the order of preference. If a property that holds for the intended referent is false for at least one

member of the contrast set, the property is added to the generated expression and the *ruled out* members are removed from the contrast set. When the algorithm has successfully eliminated all original members from the contrast set, it terminates and returns the expression generated so far. If the contrast set is non-empty after iterating over all properties, the algorithm fails.

In order to determine appropriate discriminating properties, the algorithm requires a set of interface functions to the knowledge base to get additional information, namely the *taxonomical specialization* of a given attribute, the *basic level category* of an entity's attribute, a model of the *user's knowledge*, and finally an ordered list of *preferred attributes*.

Algorithm 1 The Basic Incremental Algorithm for GRE

Require: r = intended referent; C = contrast set; P = preferred-attributes-list
Initialize: $DESC = \{\}$
for each $A_i \in P$ **do**
  $V = findBestValue(r, A_i, basicLevelValue(r, A_i))$
  if $RulesOut(\langle A_i, V\rangle) \neq nil$ **then**
    $DESC = DESC \cup \{\langle A_i, V\rangle\}$
  end if
  if $C = \{\}$ **then**
    if $\langle type, X\rangle \in DESC$ for some X **then**
      return DESC
    else
      return $DESC \cup \{\langle type, basicLevelValue(r, type)\rangle\}$
    end if
  end if
end for
return failure

Algorithm 2 findBestValue(r,A,initial-value)

if $userKnows(r, \langle, initial-value\rangle)$ **then**
  $val = initial\text{-}value$
else
  $val = null$
end if
if $(more\text{-}specific\text{-}value = moreSpecificValue(r, A, val)) \neq nil \wedge$
$(new\text{-}value = findBestValue(A, more\text{-}specific\text{-}value)) \neq nil \wedge$
$(|rulesOut(\langle A, new\text{-}value\rangle)| > |rulesOut(\langle A, val\rangle)|)$ **then**
  $val = new\text{-}value$
end if
return val

Algorithm 3 rulesOut(⟨A, V⟩)

if $V = null$ **then**
  return nil
else
  return $\{x : x \in C \wedge userKnows(x, \langle A, V\rangle) = false\}$
end if

Our approach

A robotic office assistant that is supposed to interact with its users through spoken language will have to refer to things and places in their environment. It needs to do so in a way that is intuitively understandable by humans. There are conceivably many ways in which a robot might to refer to things in the world and most of them will fail to achieve their communicative goal. Consider the following set of examples:

1. "the location at position $(X = 5.56, Y = -3.92, \theta = 0.45)$"
2. "the mug left of the plate right of the mug left of the plate"
3. "Peter's office no. 200 at the end of the corridor on the third floor of the Acme Corp. building 3 in the Acme Corp. building complex, 47 Evergreen Terrace, Calisota, Planet Earth, (...)"
4. "the area"

These referring expressions are valid descriptions of their respective referents. Still they fail to achieve their communicative goal, which is to specify the right amount of information that the hearer needs to uniquely identify the referent. First of all, robots are good at measuring exact distances, humans are not. So the robot should employ qualitative descriptions that make use of the same concepts as a human-produced utterance would. Second, specifying a referent with respect to another referent that is only identifiable relative to the first one leads to infinite recursion instead of the communicative goal. Finally, the robot might have a vast knowledge about facts and entities in the world, but it should not always try to uniquely separate the referent from all entities in the world. At the same time, it is necessary to provide enough information to distinguish the intended referent from those entities in the world that potentially distract the hearer. The following expressions *might* serve as more appropriate variants of the previous examples:

1. "the kitchen around the corner"
2. "the red mug left of the china plate"
3. "Peter's office"
4. "the large hall on the first floor"

The fact that these *might* (or *might not!*) be successful referring expressions points to the importance of knowing what the given context in a situation is. This is especially the case for a mobile robot that operates and interacts in large-scale space. It is thus an important basis to endow the robot with a spatial representation that resembles the way humans conceive of their environment. But it is not enough; the robot must also be able to determine which entities in the world might act as *potential distractors* with respect to the hearer's knowledge.

In the following sections we will describe how the ontological representation of spatio-conceptual knowledge in our robot architecture serves as a natural knowledge base for the incremental GRE algorithm. Furthermore, we will propose a method for a proper construction of the *contrast set* for large-scale space.

Proc. LangRo'2007
978-972-96895-2-9

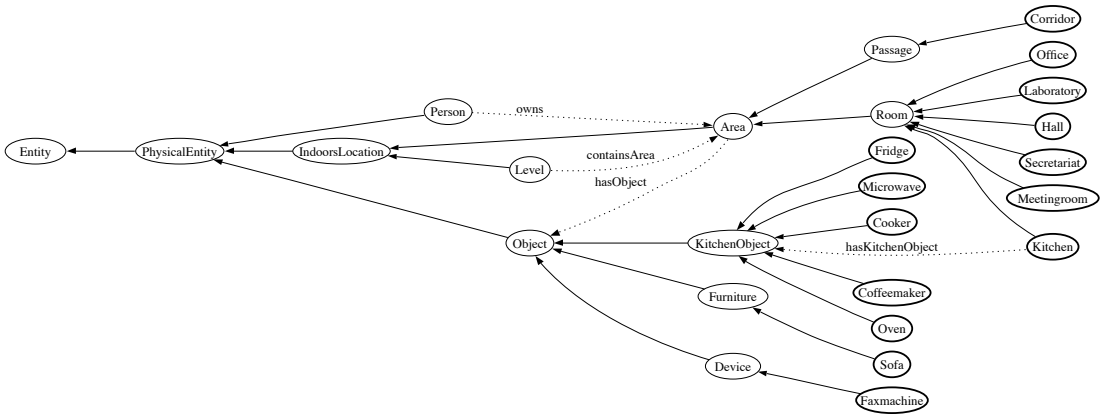


Figure 1: A subset of our office environment commonsense ontology, including relevant relations (dotted arrows) and basic level categories (thick lines).

The knowledge base

Our robotic system is endowed with a *multi-layered spatial representation*, ranging from a low-level metric map, via a topological abstraction layer, to an ontology-based conceptual map. We refer the reader to our previous publications for a more detailed account on the spatial representation (Zender and Kruijff, 2007; Zender et al., 2007). Here, we will focus on describing the relevant mechanisms for the GRE task in large-scale space.

The *conceptual map* layer consists of a Description-Logics based OWL-DL reasoner. It contains innate conceptual commonsense knowledge about an indoor office environment (Figure 1), i.e. types of spatial areas, objects and persons, and the relations that can hold between them. While the robot is exploring its environment, it populates its ontology with acquired and inferred instance knowledge.

The **instances** in the ontology are the *entities* of the world model. The **conceptual hierarchy** provides the taxonomical *type* information of the instances that the GRE algorithm requires. Furthermore, a number of concepts such as `Office`, `Kitchen`, `Corridor`, `Table`, etc. are marked as basic level categories, cf. (Brown, 1958) and (Rosch, 1978). The **relations between instances** are the *attributes* that the algorithm can use to further specify a referent. Note that the use of relations leads to a recursive call of the GRE algorithm with its well-known implications. An extension of the algorithm with heuristics to exclude reference to an already mentioned entity and to keep the recursion depth minimal can be used to cope with this issue. Currently, our approach supports the following properties (in the order of preference):

Type We represent an entity's type as the (asserted and inferred) concepts of the corresponding instance. Through ontological reasoning, we can retrieve an instance's most specific concept, its basic level category, and all the instances of a concept. Further, taxonomy traversal functions (e.g. *getSuperConcepts*,*getSubConcepts*) can provide additional type information if necessary.

Topological inclusion If the current context spans topological units at different hierarchical levels (cf. Figure 2) it is important to specify the intended referent with respect to the topological unit that contains the referent, e.g. when referring to "the kitchen on the 3rd floor", or "the table in the lab". The conceptual map represents topological position with the following relations: `hasObject(Area,Object)`, and `containsArea(Level,Area)`.

Ownership Areas in an environment are often referred to by identifying their owners, e.g. "Bob's office". In our ontology instances of `Area` can be related to a `Person` instance via the `isOwnedBy(Area,Person)` relation. The name of the person is represented as a string datatype property.

Topological connectivity Information about neighboring areas can be a good cue for identifying spatial entities, e.g. "the room next to the lab". Our system represents adjacency of topological areas in the topological layer of the robot's multi-layered map, where the information can be accessed.

Name As names are usually (locally) unique, e.g. "the Occam meeting room", or "office 120", they are definitely a highly discriminating attribute for the GRE task. However, names do not seem to be a preferred category for referring to rooms as they seldom contain more useful information than a generic NP + PP referring expression, e.g. "the meeting room on the first floor next to the large hall". On the contrary, such a generic referring expression might even bear additional useful information. Moreover, remembering the inherently artificial name for an entity might involve a higher cognitive load than processing the information encoded in a more generic referential description. For other scenarios though, such as an information desk agent at a hospital, or any other institution in which there is a specific naming scheme, such as e.g. encoding floor number and department, and numbering them in sequential order, the name feature can conceivably be

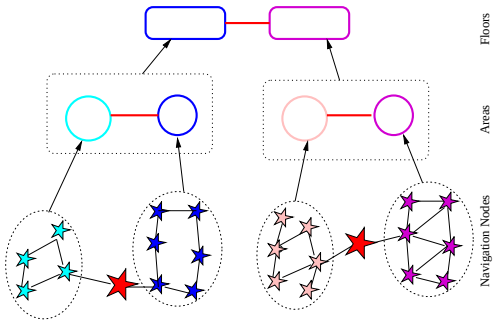


Figure 2: A topology of places, rooms and floors. Stars depict navigation nodes that denote free and reachable space for our robotic system. The set of navigation nodes is partitioned into distinct spatial areas, such as e.g. rooms. Areas in turn can belong to a floors, which are on the next level of abstraction. Using *topology traversal*, we construct an appropriate context set for the GRE task.

placed in a higher-ranking position in the preference list. In our ontology names for areas are represented as a string datatype property.

Landmarks The conceptual map contains spatial areas in the environment as well as objects found there. The information about which objects are found there can be used to further specify a spatial entity. Currently, our model only provides the information which areas contain which objects (`hasObject(Area,Object)`). The GRE algorithm can thus form expressions like "the room with the mailboxes". Since most of these objects will only be perceivable once the person is at the location of the intended referent, we assume that this attribute should only be used with a low preference. It is noteworthy that our DL-reasoner is able to categorize spatial areas on the basis of the objects that are found there (Zender and Kruijff, 2007). The knowledge about which objects are found where is thus reflected already in the type information, which is always used in the process of generating a referring expression.

Determining the appropriate contrast set

In order to successfully identify a referent it is important to determine a correct and appropriate contrast set. If the contrast set is chosen too small, the hearer might find it difficult to uniquely identify the intended referent with respect to his or her knowledge. If, on the other hand, a too large contrast set is assumed, the generated referring expression might violate *Grice's Maxims*, here the Maxim of Quality, in that it contains too much unnecessary information.

Since the contrast set is defined relative to a context set, the crucial task is hence to determine which part of the environment constitutes the current context. We claim that the context for a dialogue situated in large-scale space can be determined on the basis of a topological representation. Assuming a topological hierarchy of places, the context should include all sibling nodes of those topological units that are visited when following the search path between the current position and the intended referent (**topology traversal**). For instance, if the intended referent is an object located in the same room as the user and the robot, only local landmarks should be considered. Likewise, if the robot is to produce a referring expression to a room on a different floor, all entities on that floor and on the current floor will form the context. Using topological inclusion as the most preferred attribute will then essentially function as a pruning of the hierarchically ordered context set (if it has discriminatory power at all, that is).

In our implementation, the lowest topological level is the navigation graph. The set of navigation nodes is then partitioned into topological areas that correspond to basic spatial units, such as rooms and corridors. Our ontology additionally contains a representation for dividing areas into storeys to which they belong, cf. Figure 2. The topological unit that is considered the current position need not necessarily be the robot's and/or the hearer's physical location. We claim that our approach will also yield plausible results when used in an incremental dialogue to generate route descriptions. In that case, the most recent dialogue referent is assumed as the initial position.

Representing the user's knowledge

In the incremental algorithm the *userKnows* function is used to ensure that the algorithm does not include descriptions that the hearer does not understand and also to prevent the algorithm from ruling out members of the contrast set that are no potential distractors with respect to the user's knowledge. In our scenario, it is infeasible to fully specify the knowledge of all possible interlocutors. We therefore opt for *a priori* assuming an omniscient user. Using a dialogue model, we can explicitly mark information as not known by the user when, e.g. answering questions.

Moreover, the representation of the user's knowledge plays important role for example in the route description generation task. There, the *UserKnows* function should initially return false for any knowledge pertaining to referents that have not yet been introduced. The task of generating a route description is then redefined in terms of successively introducing new discourse referents that can then be used for the GRE task.

Natural language processing

In our system, we use a communication system for situated spoken dialogue between the robot and a user. Our implementation of the GRE algorithm collects information from the ontology that it will then represent as a Hybrid Logics Dependency Semantics (HLDS) logical form (Baldridge and Kruijff, 2002). This HLDS logical form is then processed by the OpenCCG realizer, which generates a natural language expression (Baldridge and Kruijff, 2003). The following list shows how information from the ontology is translated to HLDS. The logical form generated by the *type* attribute is the root node; logical forms representing other attributes are dependent structures of a root node.

Proc. LangRo'2007
978-972-96895-2-9

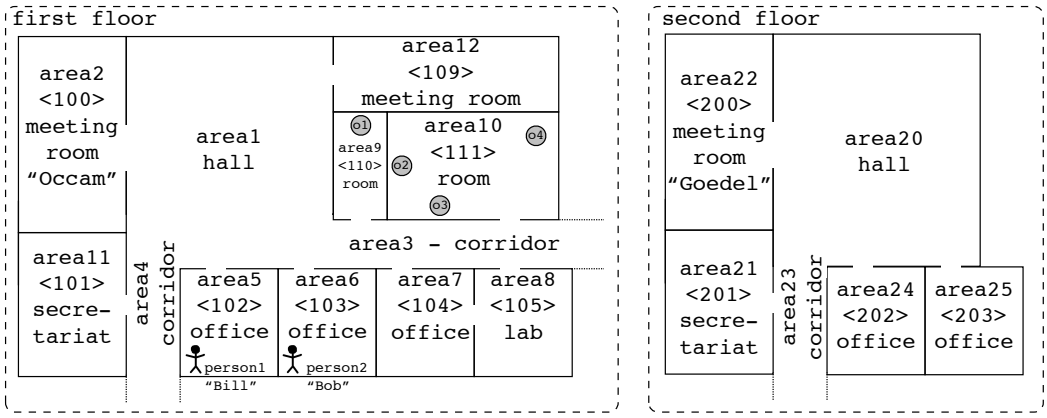


Figure 3: An example office environment knowledge base. The ontology contains several instances of areas (indicated by the `areaX` tag) on two floors, objects (`o1` is an instance of `Faxmachine`, `o2` and `o4` are instances of `Couch`, and `o3` is a `Coffemaker`), and persons (`Bill` and `Bob`).

- HLDS logical form for *type*:
  $@_{\{X:\text{entity}\}}(\langle\textbf{TYPE}\rangle$
  $\&\ \langle Delimitation\rangle\textbf{unique}$
  $\&\ \langle Number\rangle\textbf{singular})$

- HLDS logical form for *topological inclusion* (of areas):
  $@_{\{X:\text{location}\}}(\textbf{on}$
  $\&\ \langle Anchor\rangle(location\ \&\ \textbf{floor}$
  $\&\ \langle Delimitation\rangle\textbf{unique}$
  $\&\ \langle Number\rangle\textbf{singular}$
  $\&\ \langle Property\rangle(q-position\ \&\ \langle\textbf{ORD}\rangle))))$

- HLDS logical form for *topological inclusion* (of objects):
  $@_{\{X:\text{location}\}}(\textbf{in}$
  $\&\ \langle Anchor\rangle(\langle\textbf{REFERRING EXPRESSION}\rangle))$

- HLDS logical form for *topological connectivity*:
  $@_{\{X:\text{location}\}}(\textbf{next to}$
  $\&\ \langle Anchor\rangle(\langle\textbf{REFERRING EXPRESSION}\rangle))$

- HLDS logical form for *ownership*:
  $@_{\{X:\text{genOwner}\}}(person\ \&\ \langle\textbf{NAME}\rangle)$

- HLDS logical form for *name*:
  $@_{\{X:\text{identifier}\}}(loc-name\ \&\ \langle\textbf{LOCATION NAME}\rangle)$

- HLDS logical form for a number as *name*:
  $@_{\{X:\text{identifier}\}}(number\ \&\ \langle\textbf{LOCATION NUMBER}\rangle)$

- HLDS logical form for *landmark*:
  $@_{\{X:\text{accompaniment}\}}(\langle\textbf{REFERRING EXPRESSION}\rangle))$

Examples

Let us consider the example scenario depicted in Figure 3. For visualization purposes we have annotated a map sketch with the instance knowledge that is represented in the conceptual map. The knowledge base consists of a number of areas that are anchored in the topological map layer. The robot knows that the rooms in its environment are numbered (`<X>`), and that two of the meeting rooms additionally have names (`''Occam''` and `''Goedel''`). Additionally, the robot has learned the types of the areas through situated dialogue with its user (Kruijff et al., 2007). It knows about the presence of four objects, and, finally, the robot knows two persons, and in which offices they work.

The following examples are the results of applying the algorithm under varying circumstances. The initial position and the intended referent are denoted by $i=$`area`$_i$ and $r=$`area`$_r$, respectively.

(1) $i=$`area3`; $r=$`area1`;
Since initial position and target location are on the same floor, the expression "the hall" is produced.
$@_{\{area1:\text{e-location}\}}(\textbf{hall}$
$\&\ \langle Delimitation\rangle\textbf{unique}$
$\&\ \langle Number\rangle\textbf{singular})$

(2) $i=$`area3`; $r=$`area20`;
Since initial position and target location are on different floors, the expression "the hall on the second floor" is produced.
$@_{\{area20:\text{e-location}\}}(\textbf{hall}$
$\&\ \langle Delimitation\rangle\textbf{unique}$
$\&\ \langle Number\rangle\textbf{singular})\quad\&\ \langle Location\rangle(\textbf{on}$
$\&\ \langle Anchor\rangle(location\ \&\ \textbf{floor}$
$\&\ \langle Delimitation\rangle\textbf{unique}$
$\&\ \langle Number\rangle\textbf{singular}$
$\&\ \langle Property\rangle(q-position\ \&\ \textbf{2})))$

The system is able to successfully generate a referring expression (Ex. 3) for the coffee maker (`o3`), but not for any of the couches (`o2` and `o4`) because the knowledge base does not contain any information that can properly distinguish between the two.

(3) $i=$`area3`; $r=$`o3`;
The position of the intended referent is anchored in the navigation graph topology. The context set thus spans two topological layers (the navigation graph and the area layer). Hence the algorithm includes the location information when generating the referring expression "the coffee maker in the kitchen".

Ex. 3 shows a weakness of the proposed algorithm. The basic level category `Kitchen` for `area10` is inferred on the basis of the presence of a `Coffeemaker` instance. However, when generating the referring expression for the "coffee maker" this is not taken into account. It remains an issue of further research to what extent this influences the acceptability of such a referring expression.

Another observation can be made when generating a referring expression for `area4` with different initial positions. However, if a recalculated contrast set is provided as input for the recursive call to generate an embedded referring expression, we can avoid redundant attributes. Here again, we make use of our principle to determine the context on the basis of the topological hierarchy. We simply assume the position of the most recent referent as the initial position when determining the contrast set. The result of this modification leads to the result in Ex. 6

(4) $i =$`area1`; $r =$`area4`;
This configuration yields "the corridor next to the secretariat".

(5) $i =$`area20`; $r =$`area4`;
The unmodified algorithm yields "the corridor on the first floor next to the secretariat on the first floor".

(6) $i =$`area20`; $r =$`area4`;
After the modification the algorithm produces "the corridor on the first floor next to the secretariat".

Conclusions

In this paper we have presented an approach to applying the *incremental algorithm* for GRE to the domain of large-scale space, with an emphasis on its application in a mobile robot office assistant scenario. We have shown how our method of *conceptual spatial mapping* provides a knowledge base for the GRE algorithm. We have argued further that the construction of the *context* and *contrast* sets using our method for topology traversal is an important step towards generating appropriate referring expressions in large-scale space. More importantly, the same method can be used in scenarios where the robot has to provide a verbal route description from a given start position to a target location. The representation of the user's knowledge is another important parameter for the route description task where new discourse referents have to be introduced sequentially in order to allow for the generation of appropriate referring expressions.

Future work

In our current approach, the list of preferred attributes is static. Other approaches, e.g. (Kelleher and Kruijff, 2006), have shown that a dynamic ordering of attributes based on their (relative) salience yields better results. It remains an issue of future work to explore the effect and measurability of different kinds of salience (e.g. visual and discourse salience) in the context of GRE for large-scale space. A first approach could be to work with different preference lists for different types of referents (e.g. objects vs. areas).

The aforementioned approach of (Kelleher and Kruijff, 2006) provides an excellent opportunity for integrating qualitative spatial reasoning for small-scale space with the more allocentric conceptual spatial reasoning method of the approach presented in this paper. Using the method of topology traversal, the robot could conceivably produce referring expressions that incorporate entities and properties at different levels of abstraction, thus leading to a *"zooming-in-and-out"* behavior, like e.g. "the ball to the left of the box on the table in the kitchen on the third floor". Since both approaches are compatible in that they build upon on the same basic *incremental GRE algorithm*, we claim that the capabilities of our robot to refer to entities in the world can be significantly improved by combining these approaches.

References

Baldridge, J. and Kruijff, G.-J. M. (2002). Coupling CCG and hybrid logic dependency semantics. In *Proc. of the 40th Annual Meeting of the ACL*, pages 319–326, Philadelphia, PA, USA.

Baldridge, J. and Kruijff, G.-J. M. (2003). Multi-modal combinatory categorial grammmar. In *Proc. of the 10th Conference of the EACL*, Budapest, Hungary.

Brown, R. (1958). How shall a thing be called? *Psychological Review*, 65(1):14–21.

Dale, R. and Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Funakoshi, K., Watanabe, S., Kuriyama, N., and Tokunaga, T. (2004). Generation of relative referring expressions based on perceptual grouping. In *Proc. of COLING '04*, Geneva, Switzerland.

Horacek, H. (1997). An algorithm for generating referential descriptions with flexible interfaces. In *Proc. of the 35th Annual Meeting of the ACL and 8th Conf. of the EACL*, Madrid, Spain.

Kelleher, J. and Kruijff, G.-J. (2006). Incremental generation of spatial referring expressions in situated dialogue. In *In Proceedings of Coling-ACL '06*, Sydney, Australia.

Kruijff, G.-J. M., Zender, H., Jensfelt, P., and Christensen, H. I. (2007). Situated dialogue and spatial organization: What, where...and why? *International Journal of Advanced Robotic Systems*, 4(1):125–138.

Kuipers, B. J. (1977). *Representing Knowledge of Large-scale Space*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Rosch, E. (1978). Principles of categorization. In Rosch, E. and Lloyd, B., editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Zender, H., Jensfelt, P., Óscar Martínez Mozos, Kruijff, G.-J. M., and Burgard, W. (2007). An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proc. of AAAI-07*, pages 1584–1589, Vancouver, BC, Canada.

Zender, H. and Kruijff, G.-J. M. (2007). Multi-layered conceptual spatial mapping for autonomous mobile robots. In *Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems*, AAAI Spring Symposium 2007.

Maximum Entropy Statistical Modeling of Sensor Data from Robotic Systems

Daniel Burfoot and Yasuo Kuniyoshi
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{burfoot, kuniyoshi}@isi.imi.i.u-tokyo.ac.jp

Abstract

The use of probabilistic models in robotics is becoming mainstream. These models are generally designed to serve a specific purpose, such as mapping or control. This paper argues that, rather than building specialized models suitable for some specific task, roboticists should focus on the problem of building *general* statistical models. This is simply the problem of constructing a model of the sensor data, which is equivalent to compressing the sensor data. This perspective has two advantages. First, it allows strong comparisons of different techniques: models which compress the sensor data to a smaller size are better. Second, the utility of the general statistical model is potentially much broader than a model designed to serve a specific purpose. This view is motivated by the field of statistical natural language processing, where constructing a language model is a core problem for many different applications. Furthermore, the general statistical modeling view opens a new door for statistical learning due to the immense amount of data available, which justifies the use of highly complex models.

We present a technique, based on the Maximum Entropy framework, for general statistical modeling of laser range data from a mobile robot. We show that the model obtained in this way can be used to localize the robot from sensor observations, and can be used to detect changes in the environment.

Introduction

The use of probabilistic models for robotics has recently become mainstream. However, the models employed are typically intended to be used for some specific purpose, such as mapping (Thrun, 2002) or control (Wolpert et al., 2003). The idea of this paper is that it is important and useful to construct general statistical models without knowing in advance what the application of the model will be. If a good general statistical model can be obtained, it can be used for many different applications. This paper illustrates the idea in the well-known scenario where the system is a mobile robot and the data comes from laser range finders.

Current robotic systems can generate data at rates on the order of 100 Mb/s. Transmitting or storing this data would seem then to be an impossible task. However, in spite of this nominally massive data rate, it is clear that the fundamental information generation rate is much lower because the data is highly redundant. For example, each image frame contains a large amount of information about the subsequent frame. By creating a code which exploits this correspondence, we should be able to significantly reduce the code length of the data. One of the basic results of information theory tells us that in order to achieve the best possible data compression rate, we need a perfect model of the probability distribution that generated the data. Discrepancies between the real source and our model will lead to suboptimal compression rates. Thus, our goal of compressing data leads us directly to the goal of constructing a good model of the data. Conversely, we can judge the quality of our model by how well it allows us to compress the data.

Several of the ideas in this paper are inspired by the field of statistical natural language processing (SNLP). A major goal of SNLP is to create a good language model, which is simply a probability distribution $p(s)$ over sentences in the language. Though there are many interesting applications of SNLP, three important ones are speech recognition, machine translation, and text compression. By the equivalence between statistical modeling and encoding mentioned above, it is clear that the central requirement for text compression is to have a good language model. It turns out that this is a requirement for the other two applications also. The machine translation and speech recognition problems are generally formulated as finding s^* in the following equation:

$$s^* = \arg\max_s p(x|s)p(s) \tag{1}$$

Here s is a sentence, and x is either the translation of that sentence into another language (machine translation) or the spoken sound of the sentence (speech recognition). Then the system chooses s^* as either the translation of the sentence into another language, or the text version of the utterance. It is clear that for this technique to work, one must use a good language model $p(s)$.

We see that in SNLP a good language model is a widely (perhaps universally) useful tool. This is an important conceptual advance for two reasons. First, progress in language modeling can be compared easily, by using the compression rate. Second, it allows advances made in one area to immediately benefit many different applications (a similar effect to how an increase in CPU speed immediately benefits a diverse set of software applications).

In order to create good language models, one typically

first obtains a large corpus of text. Because of the internet, it is now possible to access corpuses of text that are on the order of gigabytes in size. Then a statistical model is trained using the corpus, leaving out some part to be used as test data. Finally one measures the log likelihood of the test data given the model. This value is equivalent to the theoretical optimal code length to which the test data can be compressed using a code based on the model. Several techniques exist for constructing codes based on probability distributions which yield asymptotically optimal code lengths (Witten et al., 1987).

One of the most powerful set of techniques in SNLP is based on the principle of Maximum Entropy (Berger et al., 1996; Rosenfeld, 1996). This principle, first advocated by (Jaynes, 1957), instructs us to select the statistical model with the highest entropy that satisfies the set of constraints imposed by the data. The Maximum Entropy framework is also attractive because it allows us a wide degree of liberty to define features which reflect our knowledge of the structure of the domain. For example, in SNLP one feature which is sometimes used is based on the part of speech of the previous word. In English, if the previous word is an article ("a", "an", "the") it becomes highly probable that the subsequent word is a noun. Thus a feature which can exploit such structure should be able to improve the compression rate.

This paper is essentially an attempt to transplant both the overall approach of general statistical modeling and a particular technique (MaxEnt) from SNLP to robotics. Given any robotic system one can easily define a set of models for the data produced by that system, and compare those models using the code length / log-likelihood. The key analogy here is between a corpus of text and a "corpus" of sensor data generated by a robotic system.

The most basic application of this research can be found in the direct link between modeling and compression. Compressing the large amount of sensor data generated by a system is clearly beneficial for several applications. For example in the case of remote operation it is beneficial to compress the data as it is sent from the robot to the operator. In the case of surveillance, the compression rate provides a natural mechanism by which to detect abnormalities. We also expect that, as is the case in SNLP, a good statistical model of the sensor data will be useful for a variety of applications.

The research described in this paper is superficially similar to the problem of simultaneous localization and mapping (SLAM), which has a large literature (Montemerlo et al., 2002; Dissanayake et al., 2000). However, in our work the emphasis is on modeling of data, rather than obtaining a map.

Maximum Entropy Framework

The Maximum Entropy (MaxEnt) framework describes a set of methods for modeling data. These methods have recently had great success in modeling natural language (Pietra et al., 1997; Rosenfeld, 1996). In this paper we use the MaxEnt framework to model sensorimotor data. The MaxEnt framework is appealing because it allows us wide liberty in defining features for use in constructing the model. However, the general approach to statistical modeling of sensorimotor data described in the introduction does not depend on the choice of one particular modeling tool.

Maximum Entropy as a general principle of statistical inference was first described by (Jaynes, 1957). Given a set of observations, the principle instructs us to select the maximum entropy model which is consistent with the observations. This leads to a probability distribution of the form:

$$p(x) = k \exp\big(\sum_i \lambda_i f_i(x)\big) \tag{2}$$

Where k is a normalization constant and $f_i(x)$ are a set of *feature functions*. The parameters λ_i are determined by a set of constraints given by the data:

$$E_p[f_i] = \frac{1}{N} \sum_{data} f_i(x) \tag{3}$$

In other words, we require that the empirical expectations of the feature functions be equal to the expectations induced by the model. In theory the feature functions can take any form that is thought to be useful. In practice they are often binary functions of the form:

$$f_i(x) = \begin{cases} 1 & (cp_i(x) == T) \\ 0 \end{cases} \tag{4}$$

Where $cp(x)$ is a *context predicate* which can be true or false. Context predicates can also be thought of as a set of labels which are attached to a data sample. In this paper we will be particularly interested in *conditional* Maximum Entropy estimation. Here we are interested in obtaining estimates of $p(y|H)$, where H is the history of data preceeding y. This situation is covered with the same basic procedure, except now we have more complex feature functions:

$$f_{ij}(y,H) = \begin{cases} g_i(y) & (cp_j(H) == T) \\ 0 \end{cases} \tag{5}$$

We refer to the $g_i(y)$ as *conditional functions*. Thus, a feature function f_{ij} consists of a context predicate cp_j and a conditional function g_i. The conditional functions may have different forms, which lead to different models. When y is not interpretable in terms of continuous numeric values, a common choice is $g_i(y) = \delta_{yy_i}$. The feature function only fires for a particular value of y. This technique can be used even if the values for y are numeric, but may require a large number of parameters, depending on the quantization. A model with fewer parameters but good performance can often be defined by using two functions $g_1(y) = y$ and $g_2(y) = y^2$. This corresponds to building an MaxEnt model from the first and second moments of y, which gives a Gaussian distribution.

Finding the Parameters

The main difficulty in maximum entropy estimation is to obtain the parameters of the distribution (λ_i above). There are several algorithms for this purpose (Malouf, 2002), the most well-known of which is called Generalized Iterative Scaling (Darroch and Ratcliff, 1972). The algorithms all attempt to maximize the log likelihood of the data under the model:

$$L(\Lambda) = \sum_{data} \tilde{p}(y,H) \log q_\Lambda(y|H) \quad (6)$$

Where $\tilde{p}(y,H)$ is the empirical distribution and $q_\Lambda(y|H)$ is the model with parameters Λ. An equivalent formulation of the problem is to find the set of parameters such that:

$$\sum_{data} q_\Lambda(y|H)\tilde{p}(H)f_{ij}(y,H) = \sum_{data} \tilde{p}(y,H)f_{ij}(y,H) \quad (7)$$

The equivalence between these two formulations can be understood as follows. For probability distributions of the form (2), the set of parameters which maximizes the likelihood of the data and the set of parameters which produce the maximum entropy distribution while respecting the constraints produced by the data are identical (Berger et al., 1996).

In this work we use the standard Generalized Iterative Scaling (GIS) algorithm. This technique finds the optimal parameters λ_{ij} by performing updates of the form:

$$\Delta\lambda_{ij} = \log\left(\frac{E_{\tilde{p}}[f_{ij}]}{E_{q_\Lambda}[f_{ij}]}\right)^{\frac{1}{C}} \quad (8)$$

Where $E_{\tilde{p}}$ denotes empirical expectations (RHS of Eqn. 7) and E_{q_Λ} denotes expectations under the current model (LHS of Eqn. 7). The value C is a constant such that $\sum_{ij} f_{ij}(x) = C$ for each element of the training data, which is required for the proof of convergence. This can be easily satisfied by the use of dummy features, but if the value of C is large, it can slow down convergence. The log likelihood is a convex function of the parameters λ_{ij}, so this technique is not affected by local minima.

Statistical Modeling Techniques

Our goal in this research is to give an example of how the general statistical modeling paradigm can be applied to robotics. For this reason, we use a simple and well-known application area: a mobile robot navigating in an unknown environment, collecting sensor data from a set of laser range finders. The environment used in shown in Figure 1. The robot's control system consists of a simple Braitenberg architecture. This basically tells the robot that if an obstacle is detected on the left side, the robot should turn right. Surprisingly, this very simple behavior leads to good obstacle avoidance properties (Braitenberg, 1984).

In this section we describe two different approaches to statistical modeling of the robot data. The first one is based on attempting to model probability distribution for sensor readings based on features of the history such as the previous sensor readings, and the location of the robot. In the second approach, we transform the sensor readings into a new event space, create a statistical model in the new space, and then transform back to predict the sensor readings.

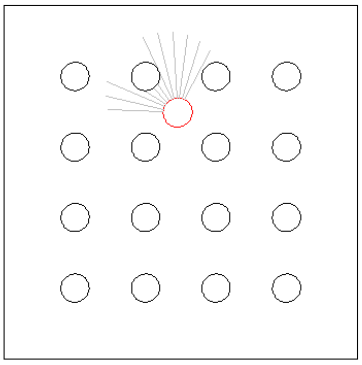


Figure 1: Map A, used to generate training data. The robot is shown in red, with grey lines indicating sensor rays.

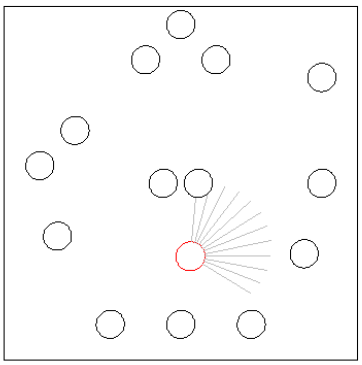


Figure 2: Map B, used to test for environmental change detection.

Direct Statistical Modeling of Sensor Readings

In this section we describe techniques for constructing probability distributions $p(y|H)$ over sensor readings y given the history H. In principle, the history can include any information which is available to the robot at the time of prediction. In the following, we notate s_k^t as the reading of sensor k on timestep t, whereas $\mathbf{s^t}$ denotes the full set of sensor readings on timestep t Both the previous sensor reading $\mathbf{s}^{t-1}$ and the location $\mathbf{r}$ are understood to be included in the history H. We make a distinction between next sensor reading y which is the object of prediction and past readings $\mathbf{s}$ which can be used for the prediction. Note also that we are building separate statistical models for each of the 12 sensors. Some Trackers assign the same context predicate to all sensors, while others assign different predicates to each (see Table 1).

To use the Maximum Entropy framework, one must define a set of feature functions that will be useful in predicting sensor readings. In statistical natural language processing, these feature functions typically refer to the words preceeding the target word in a sentence. For example, if the preceding word is "the", there is a high probability that the subsequent word will be a noun. In this work we define feature functions which refer to various functions of the previous sensor readings and the robot location.

In the definitions below we refer also to the conditional functions $g_i(y)$. In the experiments described below we use two sets of conditional functions, "delta" and "numeric". The delta conditional functions are of the form $g_i(y) = \delta(y - y_i)$, where y_i range over all possible sensor readings. The numeric conditional functions are $g_1(y) = y$, $g_2(y) = y^2$. If used alone, these two functions suffice to determine a Gaussian probability distribution. However, we observed in the course of experimentation that very often the empirical distribution corresponding to a particular feature function would include a roughly Gaussian set of values in an intermediate range, and additionally a significant number of values at the maximum sensor reading. For this reason we added a third conditional function $g_3(y) = \delta(y - y_{max})$ to the numeric set, where y_{max} is the maximum sensor reading. The numeric conditional functions use fewer parameters than the delta conditional functions (3 vs 64), so we use the latter only in conjunction with feature functions that generate a small number of predicates (the total number of parameters is the number of conditional functions times the number of predicates).

We refer to a package of feature functions, conditional functions, and associated logic as a "Tracker". We have defined several different types of Trackers, which are described below. They have different properties as shown in Table 1. This table shows the approximate number of predicates that can be produced by the Tracker, the number of predicates generated per step, the type of conditional functions $g_i(y)$ used for the Tracker, and whether the Tracker is location-dependent or sensor-specific.

Previous Value Tracker (PVT) - use the value of the sensor on the previous timestep (s_k^{t-1}) to predict the next value:

$$v_k = s_k^{t-1} \tag{9}$$

$$f_{ij}(y,H) = \begin{cases} g_i(y) & (V_j \le v_k < V_{j+1}) \\ 0 \end{cases} \tag{10}$$

Since there are 64 values the reading can take on, there are 64 possible context predicates for each sensor ($V = \{0,1,2\ldots63\}$) At each time step there can be only one active context predicate of this type per sensor.

Neighbor Previous Value Tracker (NPVT) - use the neighboring sensor readings on the previous step, that is $s_{k\pm1}^{t-1}$. As with the Previous Value Tracker, there are 64 possible values. The intuition behind this tracker is that sensor readings can often be discontinuous as the path of the sensor ray suddenly connects with an obstacle. This could cause very bad prediction breakdowns, if the prediction system assumed that the sensor reading doesn't change rapidly. Typically, however, if a discontinuity is about to occur then there should be a large gap between the previous value of the sensor and the previous value of its neighbor.

Obstacle Tracker (OBST) - Here we use a simple geometric obstacle position estimator (denoted *OPE*) to estimate the position of an obstacle from the previous sensor reading $\mathbf{s}^{t-1}$. This function assumes the obstacles are circular and of known radius. The obstacle position estimator only fires if there are two or more adjacent sensor readings that report a non-maximum value.

$$v = OPE(\mathbf{s}^{t-1}) \tag{11}$$

$$f_{ij}(y,H) = \begin{cases} g_i(y) & (v \in R_j^{OBST}) \\ 0 \end{cases} \tag{12}$$

This requires a number of context predicates which is depends on how many regions R_j^{OBST} we use. In the following experiments, we use a quantization which gives about 1000 possible locations. Note that the locations are relative to the robot's position. This sensor will produce multiple context predicates if there are multiple obstacles detected.

Basis Function Tracker (BFT(n)) - given the previous full sensor reading $\mathbf{s}^{t-1}$ and a vector ϕ, use the dot product to define a set of feature functions:

$$v = \phi \cdot \mathbf{s}^{t-1} \tag{13}$$

$$f_{ij}(y,H) = \begin{cases} g_i(y) & (V_j \le v < V_{j+1}) \\ 0 \end{cases} \tag{14}$$

In other words we take a weighted sum of the all the sensor readings from the previous step to generate the context predicate. The question now is how to find a good choice for ϕ. The intuition behind this concept is that if a good set of basis functions is used, then one of them might respond strongly to a particular configuration of obstacles that is highly informative. In the following we use Legendre polynomials to define ϕ:

$$\phi_k^n = P_n(x_k) \tag{15}$$

Where $P_n(x)$ is the nth Legendre polynomial and the x_k are 12 equally spaced points on the interval $[-1,1]$ inclusive. The notation BFT(n) refers to a feature function defined using ϕ^n.

Simple Map Tracker (SMT) - This Tracker uses a feature function that depends only on the robot's current location, and not on its orientation. The rationale here is that the number of predicates will be prohibitively large if we use a feature function which depends on the orientation as well as the pose. Furthermore, because of the way the robot moves through the environment, it is often the case that its position in the map determines its orientation as well. For example, when the robot is located in the narrow

Proc. LangRo'2007
978-972-96895-2-9

Tracker	# Predicates	$g_i(y)$ Type	Sensor-Specific	Loc. Dep.?	Predicates per Step
PVT	64	Delta	Yes	No	1
NPVT	128	Delta	Yes	No	1-2
OBST	∼1000	Numeric	No	No	0-3
BFT(n)	∼100	Delta	Yes	No	1
SMT	4096	Numeric	No	Yes	1
SPT	∼40000	Numeric	Yes	Yes	1
REXT(n)	4096	Numeric	Yes	Yes	1

Table 1: Tracker information.

space between two obstacles, it is typically not pointed toward the obstacles (sometimes this is not the case, due to the random initialization of the robot's position). We use a quantization of the map that leads to 4096 regions R_j^{SMT}.

$$v = \mathbf{r}_{x,y} \tag{16}$$

$$f_{ij}(y,H) = \begin{cases} g_i(y) & (v \in R_j^{SMT}) \\ 0 \end{cases} \tag{17}$$

Sensor Position Tracker (SPT) - use a feature function based on the current position and direction of the sensor base:

$$v = \mathbf{r}_k \tag{18}$$

$$f_{ij}(y,H) = \begin{cases} g_i(y) & (v \in R_j^{SPT}) \\ 0 \end{cases} \tag{19}$$

Here $\mathbf{r}_k$ is distinct from the robot's location $\mathbf{r}$ because the former refers to a sensor's position on the external surface of the robot, while the latter refers to the robot's center. More importantly the $\mathbf{r}_k$ have different angular components. We assume knowledge of the relationship between the robot pose and the pose of each sensor. This Tracker should be very informative - a given sensor should always respond with the same reading at a given location. However, there are too many possible poses to create feature functions that respond exactly to every one of them. In other words, we need to coarsely quantize the data in order to limit the number of context predicates required. In these experiments we use a quantization leading to approximately 40,000 regions R_j^{SPT}.

Ray Extension Tracker (REXT) - use a set of feature functions based on the grid regions which the sensor ray goes through. Thus, we define a set of feature functions f_{ijn} which are true if the nth sensor reading for a timestep is within a region R_j. The function $EXT(\mathbf{r_k}, n)$ denotes extension: it calculates the position in world coordinates corresponding to the nth sensor reading. To limit the number of predicates, we use $n \in \{16, 32, 48, 64\}$.

$$v_{kn} = EXT(\mathbf{r_k}, n) \tag{20}$$

$$f_{ijn}(y,H) = \begin{cases} g_i(y) & (v_{kn} \in R_j^{REXT}) \\ 0 \end{cases} \tag{21}$$

Again, the number of context predicates is determined by the size of the regions R_j^{REXT}. For this Tracker we use 4096 regions. At each timestep, this Tracker produces 4 context predicates for each sensor.

Statistical Modeling in the Transition Space

In the previous section, we were interested in making predictions about probability distributions of the sensor readings $p(y|H)$. In this section, we use a different approach. Here, we transform the sensor data into a new space and create a statistical model of the data in the new space. Then, in order to predict sensor readings, we transform predictions made by the model back into the sensor space.

The new event space classifies sensor ray transitions $X \rightarrow X'$ as either STOP or PASS. The X locations are determined by the tracing the path taken by the sensor ray. In other words, for each possible sensor reading we find the position that reading corresponds to in the real world. For each timestep, this method generates a large number of PASS transitions (because the sensor ray typically passes through several positions before being stopped) and possibly one STOP transition, if an object was detected by the sensor. Thus we are interested in constructing probability distributions of the form $p(STOP|X,X')$.

The same MaxEnt framework is used, but here we apply a much smaller number of feature functions. The feature functions correspond to obstacles of known radius, at various positions. Also, in this case we have only two outcome values (PASS or STOP) as opposed to the 64 sensor reading values in the previous section.

$$cp_{ij}(X,X') = \begin{array}{l} R_{min} < D(X,P_{ij}) < R_{max} \\ D(X',P_{ij}) < D(X,P_{ij}) \end{array} \tag{22}$$

$$f_{ij}(y,X,X') = \begin{cases} 1 & (y == STOP, \quad cp_{ij}(X,X') == T) \\ 0 \end{cases} \tag{23}$$

Where $D(\ldots)$ is Euclidean distance. Thus, the feature function fires if the first point of the transition is within an annular ring of inner radius R_{min} and outer radius R_{max}, and the second point is closer to the center point P_{ij} than the first point. This function hopefully captures the notion of a sensor ray being stopped by an obstacle centered at P_{ij}. Here we assume that the obstacle radii are approximately known.

This method gives us a model for probabilities $p(STOP|X,X')$. In order to predict the sensor readings, we must transform these probabilities into probabilities for sensor readings $p(y|H)$ as in the above section. This can be done by the following simple recursive process.

$$T_0 = 1 \tag{24}$$

$$p(y = Y_i|H) = T_i \cdot p(STOP|X_i, X_{i+1}) \tag{25}$$

$$T_{i+1} = T_i - P(y = Y_i|H) \tag{26}$$

Here, T_i represents the total probability that the sensor has not yet encountered a stopping point. If the stopping probabilities were identical and equal to γ for every transition, then this would just be an exponential distribution with parameter γ. This technique captures the fact that if a sensor ray goes through the vicinity of two obstacles, each of which have equal probability of stopping it, then a sensor reading corresponding to the closer obstacle has a greater probability.

In our experiments, we used a quantization giving ten thousand possible obstacle locations (and the same number of feature functions). For completeness we also used several feature functions corresponding to walls, but as we are primarily interested in inferring obstacle locations we skip over the wall feature functions.

Experimental Results

The robot's sensor apparatus consists of 12 laser range finders. They are arrayed on the front face of robot. For the purposes of this experiment, we assume that the sensors report measurements with 6 bits of precision on each timestep, for a total of 72 bits of data per timestep.

The robot runs through trajectories which are 150 timesteps long. Thus each trajectory generates 10,800 bits of data. The training data consists of 500 trajectories for a total of 5,400,000 bits and the test is uses 100 trajectories for a total of 1,080,000 bits. We will be particularly interested in the log likelihood of the model on the test data, so we will often compare values achieved by different models to the 1080K value. We ran two sets of experiments, one of which was noise-free and one of which applied noise to the sensor values. The noise was applied by adding Gaussian random values to the sensor reading. However, if after applying the noise the distance to an obstacle was still greater than the sensor range, then the sensor reported the maximum 6-bit reading. In other words, the noise never caused the sensor to report a non-maximum value if the there was no obstacle within range. The standard deviation of the noise was one tenth of the robot's radius.

Our first goal is to determine which combination of Trackers give the best performance, measured by log likelihood on the test data. If we use too many Trackers, we will get bad performance due to overfitting. It is also possible that some of the Trackers are simply ill-suited to the task at hand.

By inspection during the course of experimentation, we observed that the Previous Value Tracker (PVT) gives the best compression rate when used alone. So, the first result reports the performance of various Trackers coupled with the PVT. Table 2 shows the results from the training phase. We report the negative log-likelihood of the model on the training data for the noisy case and the noise-free case. We also show the number of GIS iterations and time required for convergence. For compactness we show only the two Basis Function Trackers that gave the best results. Somewhat surprisingly, the Trackers based on sensor readings from the previous timestep (OBST, NPVT, and BFT(n)) give better compression rates than those based on the robot's location (SMT, SPT, REXT(n)). This may be because the previous sensor reading-based Trackers require fewer parameters. It may be that the location based Trackers can achieve good performance, but only if a larger amount of training data is used.

Tracker	Log-Like (Kbits) Noise Free	Log-Like (Kbits) Sensor Noise
REXT(16)	224	321
REXT(32)	208	312
REXT(48)	209	312
REXT(64)	215	318
OBST	210	317
SPT	261	360
NPVT	173	302
SMT	218	322
BFT(2)	194	313
BFT(4)	197	313
PVT(alone)	222	325
Trans-Space	117	293
Flat	1080	1080

Table 3: Negative Log-Likelihood achieved on the test data by combining various Trackers with PVT.

From these results (confirmed by results on the test data below) it appears that the best model can be constructed by using the PVT and the Neighbor Previous Value Tracker. We now attempt a third test in which we use three Trackers, these two combined with the others. The results for the third test are shown in Table 4. The best third Tracker appears to be the 4th Basis Function Tracker, BFT(4). Observing that the improvement in code length over the previous combination is relatively modest (165 Kbits vs 173 Kbits) we conclude that not much further progress is possible given the amount of data we are using.

Also in Table 2 is shown the results for the Transition Space model. We see that this technique performs better than the Tracker-based models.

Application to Pose Estimation

In this section we describe how the MaxEnt model developed above can be used in conjunction with a particle filter to estimate the robot's pose as it moves through the environment. In order to generate the statistical model, we assumed that knowledge of the robot's location was available. Now we show that once we have the model, we can

Proc. LangRo'2007
978-972-96895-2-9

Tracker	Train Data LL Noise Free	Train Data LL Sensor Noise	GIS Iterations	GIS time (s)
REXT(16)	680	1026	1607	2770
REXT(32)	668	1016	1475	2648
REXT(48)	669	1014	1309	2589
REXT(64)	684	1028	1161	2424
OBST	715	1064	755	8747
SPT	685	1024	1397	3619
NPVT	544	969	509	772
SMT	707	1053	1108	2055
BFT(2)	629	1021	302	485
BFT(4)	648	1033	299	526
Flat	5400	5400	-	-
Trans-Space	61	184	863	592
Flat (TS)	8936	8899	-	-

Table 2: Training information for various Trackers combined with Previous Value Tracker. "Flat" is the log-likelihood / codelength of the training data that would result from making uniform predictions of sensor readings. Also shown is analogous information for the Transition Space model. Note that the flat codelength for the Transition Space model is different from the Tracker models, because we are modeling a different event space.

Tracker	Log-Like (Kbits) Noise Free	Log-Like (Kbits) Sensor Noise
REXT(16)	192	308
REXT(32)	178	302
REXT(48)	181	301
REXT(64)	185	304
OBST	169	299
SPT	245	340
SMT	180	306
BFT(2)	166	301
BFT(4)	165	299
Flat	1080	1080

Table 4: Negative Log-Likelihood achieved on the test data by combining various Trackers with PVT and NPVT.

use it to find the robot's position.

For conciseness we give only the briefest possible description of the particle filter. The interested reader is referred to the tutorials (Arulampalam et al., 2002; Rekleitis, 2004). This technique allows us to estimate some quantity of interest, such as robot location, from probability estimates obtained with a statistical model. The algorithm represents the PDF of some variable of interest as a weighted sum of a large number of particles. The particle filter performs the following operations:

- Prediction step - update the positions of the particles by sampling from the robot's motion model.
- Update step - use the information from a newly available sensor reading to update the particle weights.
- Resample step - if the total weight is densely packed into a small set of particles, resample by dropping low-weight particles and adding copies of high-weight particles.

The main requirements for using the particle filter are a good motion model for the robot and a good observation model to use for the update step. The update step changes the probability distribution to reflect a newly available sensor reading $\mathbf{s}$ by updating the weights w_i of the particles:

$$w_i' = w_i * p(\mathbf{s}|\mathbf{r}_i) \tag{27}$$

Thus if a particle representing a position $\mathbf{r}_i$ gives a very unlikely prediction of the sensor reading $\mathbf{s}$ its weight will be greatly reduced. This observation model $p(\mathbf{s}|\mathbf{r}_i)$ is exactly the object we developed in the previous section. Unfortunately, however, we cannot use the best model as determined by the compression rate, because these Trackers do not use location information. In the following experiments we test two models using the SMT and the REXT(32), as well as the Transition Space model.

An additional requirement for using the particle filter is a motion model for the robot. This is a probability distribution over subsequent states given the previous state and the motor command. The point is that we don't assume that the motion model is perfect, but rather subject to a certain amount of noise (if there were no noise, we could use the motion model to obtain the exact location of the robot at every step). In this work, we do not consider the motor commands issued by the control system, but rather assume that the control system reports a noisy estimate of the robot's motion at each step. We separate this into linear noise applied to the robot's (x,y) position and rotational noise applied to the robot's orientation (θ). We assume that these are both zero-mean Gaussian with $\sigma = .003$. For comparison the robot's radius is .042.

In initial experiments we noticed one problem that often occurs is that while the particle filter produces good position estimates most of the time, occassionally it completely loses the position of the robot. In this case, the particles become concentrated in a location far away from

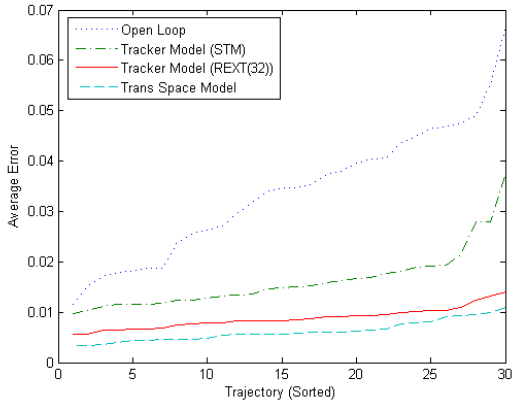


Figure 3: Average position error on particle filter localization experiments. This plot shows the average position error over an entire trajectory. The results are sorted so that the highest error trajectory for each method is shown on the right.

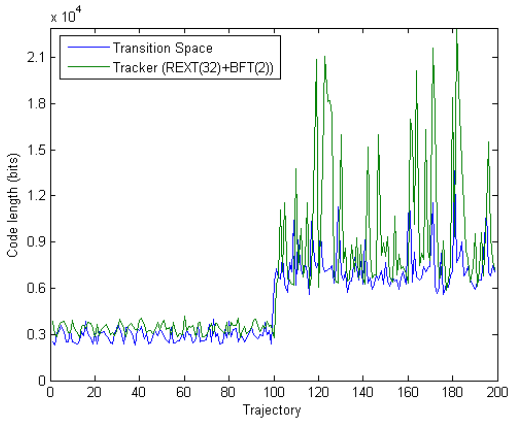


Figure 4: Detecting environmental change. After the 100th trajectory, the environment is changed from map A to map B.

the robot's actual position. This typically occurs when the robot goes through a region for which the model has insufficient data, and so produces bad probability estimates. It is then very difficult for the system to re-acquire the robot position. In contrast, the open-loop particle filter (one in which the update step is skipped) creates a wide cloud of particles, some of which almost always overlap with the robot's real position.

We developed a slightly modified version of the particle filter that exploits this phenomenon. There are some number of "hard" particles and some number of "soft" particles. All particles are updated normally using the prediction and update rules. However, only the low-weight hard particles are discarded during the resampling step. The soft particles remain, regardless of how low their weights drop. We observed that this has the following effect: sometimes the system builds up a large amount of error, but after a few steps the error suddenly drops dramatically. This can be understood as follows: the system first incorrectly assigns a large weight to some bad particles, but keeps the soft particles. It then becomes clear that the bad particles are inconsistent, (because they are making bad predictions), so weight is transferred back to some of the soft particles that are making good predictions.

In Figure 3 we show the average Euclidean distance error between the particle filter's estimate of the robot location and the real position, where the average is taken over a single trajectory. The results from 30 trajectories are sorted for the sake of comparison (the highest-error trajectory for a given model appears on the right, the lowest-error on the left). We used two Tracker based models, one using the SMT and another using the REXT(32). For these two, 1000 "soft" and 1000 "hard" particles were used. We also show the results for the Transition Space model. This model is quite computationally expensive, so we used only 100 hard particles and no soft particles in this case. However, good results can be obtained nonetheless, as shown. For comparison we show also show the error resulting from an open-loop particle filter with 1000 particles.

Application to Environmental Change Detection

In this section we describe a method for using the statistical models of sensor data developed in the previous sections for the purpose of detecting changes in the environment. The intuition here is that the probability models developed during the training phase are well-suited to the environments which were used to generate the training data. Thus if the environment is changed, the models should "detect" that change by reporting a high code length for the data from the new environment.

To test this idea, we looked at the change in codelength resulting from a change in the environment. We generated 200 trajectories. The first 100 were from the original map A shown in Figure 1, the data from which was used to train the models. The second 100 trajectories were from a different map, Figure 2. We graphed the codelength assigned to each trajectory. The results are shown in Figure 4. Evidently the technique works very well, as the code length immediately spikes when the environment changes.

However, using the best model (PVT+NPVT+BFT(4)) as indicated by the compression rates achieved in the previous section did not give perfectly clear results. This can be understood because the Trackers that were selected used primarily local information about the structure of the sensor readings. The results shown in green are generated using a combination of the REXT(32) and the BFT(2) Trackers. Also shown are the results for the Transition Space model.

Conclusion

This paper describes an approach to robotics based on the creation of general statistical models to describe the data generated by the system. Motivated by work in the field of statistical natural language processing, we attempt to build statistical models of data generated by the system. The

Proc. LangRo'2007
978-972-96895-2-9

goal is simply to find the model which achieves the best log likelihood for the data, which is equivalent to compressing the data. Because the performance of a model is graded by one number, we are able to make progress in a clear direction. A major hypothesis of this approach is that the general statistical model produced in this way will be useful for practical applications. This hypothesis is supported by the preliminary results given in this paper, as well as by work in SNLP where improvements to the language model result in increased performance in machine translation and speech recognition applications. However, we also saw that for some applications, because the different modeling techniques worked in different ways, it was not the model which produced the best compression that was the most useful.

The result of this paper showed that a general statistical tool, the Maximum Entropy framework, can be used for the problem of understanding sensor data from a mobile robot. This contrasts to the standard approach of using a specialized statistical model, such as one based on mapping. The key advantage of the general perspective is that it can be used in many other situations. If we wanted to use visual data instead of range data, it should be quite straightforward to define a new set of context predicates which respond to visual data. For example, we could define a set of predicates which respond to the presence of lines, corners, or other shapes in images.

One powerful idea in computational neuroscience is that the brain uses essentially only one learning algorithm for all of its activities (Quartz and Sejnowski, 1997). A weaker, but perhaps more plausible, version of this idea is that the brain uses a small number of basic algorithms, and all cognitive abilities arise from interactions of this small set. If this idea is true, it is a powerful argument for the general statistical modeling approach. Instead of using separate algorithms for object recognition, mapping, dextrous control, language, vision, and all the other aspects of cognition, the general approach simply considers data as a generic mass of bits in which structure must be found. This view is certainly valid from the perspective of individual neurons, which can hardly be said to "know" whether the data impinging on their synapses is visual, aural, tactile, or olfactory. Following this line of reasoning we are led to the conclusion that language scientists and roboticists fundamentally seek to understand the same thing: the process by which the brain extracts structure and meaning from data. This conclusion is supported by the results of this work, where a technique which was originally used for statistical natural language processing turned out to be useful for modeling sensory data from a mobile robot.

It can be argued that most subfields of artificial intelligence are fundamentally concerned with statistical modeling. Computer vision, robotics, planning, natural language processing, and many other facets of AI can be formulated as statistical problems. Historically, the field of statistics has operated in regimes where the data samples are relatively few in number and statistically independent. One of the great lessons of statistical learning theory is that when the data is limited, the model built from the data must be simple in order to avoid overfitting. In the VC theory (Vapnik, 1998), model complexity is quantified by the VC dimension; in MDL theory (Rissanen, 1978), it is quantified by the code length required to represent the model. So, a standard problem in statistics would be to fit a Gaussian model to a set of 1000 data points - a small amount of data, and a simple model to describe it. When we enter the regime of data rates on the order of 100 Mb/s, the situation is entirely different. The vast amounts of data available justify the use of enormously more complex models. Traditional application-oriented research in artificial intelligence ignores the vast bulk of data available, preferring instead to focus on a small subset of data for which labels or ground truth can be obtained. In doing so the research is necessarily limited in the complexity of the models it can justifiably create.

While this study is only a first step, some important principles were observed. The first is that even simple methods can yield important improvements. This can be seen as an illustration of the "cheap design" principle of (Pfeifer and Scheier, 1999). In our case we observed that predicting the sensor values based solely on the previous value yielded a decrease in code length of about 70% on the noisy data. Also, an important tradeoff exists between the fineness of the model and the amount of data which is available. Thus we saw that features based on the specific pose of the robot, which should give very strong predictions, did not succeed because an insufficient amount of data was available. However, it should be noted that simply increasing the amount of data is not a cure-all. Using more data will cause the learning algorithm to run more slowly. Depending on the scaling properties of the learning algorithm used, this may result in much greater computational effort for very minor performance gains. In order to achieve good results, we must not only use more data, but also discover ways to handle the data more efficiently.

Acknowledgements

The authors would like to thank Max Lungarella for developing the robot simulator, and for many interesting discussions. DB is supported by a Monbugakakusho scholarship from the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT).

References

Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Berger, A. L., Pietra, S. D., and Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, Mass.

Darroch, J. N. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

Dissanayake, G., Durrant-Whyte, H. F., and Bailey, T. (2000). A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *Proceedings of the 2000 IEEE international Conference on Robotics & Automation (ICRA)*, pages 1009–1014.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Phys. Rev.*, 108(2):171–190.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Sixth Conf. on Natural Language Learning*, pages 49–55.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). Fastslam: a factored solution to the simultaneous localization and mapping problem. In *Eighteenth national conference on Artificial intelligence*, pages 593–598, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. Cambridge, MA: MIT Press.

Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

Quartz, S. R. and Sejnowski, T. J. (1997). The neural basis of cognitive development: a constructivist manifesto. *Behavioral and Brain Sciences*, 20(4):537–596.

Rekleitis, I. M. (2004). A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10:187– 228.

Thrun, S. (2002). Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium.* Morgan Kaufmann.

Vapnik, V. (1998). *The Nature of Statistical Learning Theory*. Springer.

Witten, I. H., Neal, R. M., and Cleary, J. G. (1987). Arithmetic coding for data compression. *Commun. ACM*, 30(6):520–540.

Wolpert, D. M., Doya, K., and Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society London B*, 358:593–602.

Acquisition and Exchange of Knowledge From Real to Virtual Embodiment

Joerg Irran, Gregor Sieber, Marcin Skowron, Brigitte Krenn

Austrian Research Institute for Artificial Intelligence
Freyung 6/6, 1010 Vienna, Austria
joerg.irran@ofai.at

Abstract

Today's computer power enables us to create software agents that can process large amounts of data in very short time. Higher level cognitive processes, however, still remain the domain of humans. The aim of our research is to combine the power of both sides, to realize virtual agents that provide capable assistance to their users. In our approach we do not attempt to mimic human cognition. Rather, we enable the agents to learn via self-experience, from positive and negative feedback by the user, and from communication with other agents of their kind using grounded and agreed upon symbols. The design of the agents is inspired by insights from embodied cognition - in particular from affordance-based robotics - that are transferred to a virtual context.

The Rascalli Objective

In the RASCALLI (Responsive Artificial Situated Agents that Live and Learn on the Internet) project, we aim at virtual agents, the Rascalli, that are capable of autonomously exploring the Internet, and of communicating with the user and with each other. The agents have to deal with an environment that is constituted of data including strings of written language, markup tags, audio and image files, log-files of user activities, etc.

For an agent to explore its inherently dynamic environment according to the user's interests it needs to gain a certain degree of autonomous and flexible behaviour. To achieve this, we transfer insights from affordance-based research initiated by Gibson (Gibson 1986) and from affordance-based robotics research (Kintzler et al. 2007) to the design of the virtual agents and their environmental framework. Thus we develop agents whose behaviour is biologically inspired and who gain knowledge - clearly different from human knowledge, but grounded on the theory of human use of affordances - which amongst others forms the basis to realize communication between agents, and in the long run between agent and user.

Affordance-based Approach

Following the affordance approach, the environment is not perceived in terms of object or structure descriptions, but in an action oriented way, based on previously made experiences by the agent. The focus lies on what can be done with the structures in the environment - what they afford in relation to the acting agent. Thereby affordances describe the relation between the agent's capabilities and its environment. For example, an action *close fingers* applied to a small object will result in an outcome *gripped* and therefore the affordance *grippability* exists between the agent and a small object of a certain width. Representing knowledge in that manner offers a basis for flexibility in performing tasks, e.g. if a human being wants to hammer a nail into a wall - has therefore a desired outcome in mind - and no hammer is currently present, she is flexible in finding objects that can be used to hammer a nail even if they do not have the body structure of a hammer. Solving such a task involves experience fragments about wall structures and materials, *liftability* of objects, how to control one's own body, etc. that are combined and used for this particular situation. In humans this experience fragments are gained via a life-long interaction and learning process, rooted in infancy and childhood.

To gain the mentioned flexibility in solving tasks requires that the represented interaction-based knowledge is at a level of abstraction that allows generalisation and dealing with novel situations. The size of this experience base of an agent correlates with the agent's flexibility to adapt to different situations. Exploring the environment, the own interaction capabilities and outcomes of applying actions on the environment provide an experience base for further more purposeful interactions. In humans and also animals this is equivalent to play and exploration.

Transferring this biologically inspired interaction-based approach to the Rascalli agents enables the agents to gain knowledge about what they can do based on their own interaction capabilities and their own perception. This provides the basis for the required autonomous und flexible behaviour. As a very simple example, flexibility is required if a search engine or a database is unreachable. In this situation the virtual agent can choose a different one by knowing that the functionality and outcome of an alternative resource is similar. Flexibility is also required to adapt to different users and their interests – to know what types of output based on given input are appreciated by an individual user and what tools can be applied to provide the desired output.

Virtual Agents Inspired by Robotics

To realize an affordance based architecture for virtual environments, a virtual embodiment for our agents is created. They are equipped with a collection of sensor

channels geared towards the particular environment, and a set of specialized software tools (actions) through which they interact with the environment. The outcomes of the tools (of applying actions on the environment) are again treated as an input to the agent and perceived via the sensors channels in a way similar to a robot's perception of the consequences of an action application.

Depending on the developmental stage of the agent, tools are selected and executed arbitrarily, motivated by drives, or deliberately chosen based on the given input and previously made experience. Tools can be cascaded, whereby an output of a tool is part of the input of the succeeding tool. In this manner action chains are realized. After tool execution the environment, which amongst others contains the consequences of the tool application as well as user feedback and internal states of the agent, is again perceived. For each tool application the perceptions of the pre-application (I) and the post-application (O) phase are stored, over time leading to tool-specific (T) application spaces containing all the episodes experienced with the individual tools. By finding similarities and deriving representative descriptions from the individual episodes of an application space a generalization process takes place (Irran et al. 2006). These I-T-O (input-tool-outcome) triplets form the agent's experience base which enables the agent to gain representations of general input situations and desired outcomes of tool applications. It allows the agent to act more purposefully on future input. Since the episodes related with each tool application include user input and feedback, the agent does not only know what it can do, but also what is appreciated by the user and what is not. This together with the agent's drive to get positive feedback from the user forms the basis for learning successful interactions in its environment.

The Role of Communication

To enrich the capabilities of the Rascalli agents they are equipped with symbol acquiring and communication mechanisms. This allows agents to share their experiences obtained through interaction with the environment. Each agent has its own experience base comprising generalizations over the outcomes (O) of action applications, and generalizations of the input types (I) tools (T) can apply to. Due to the differences in their experience bases a negotiation process between Rascalli takes place establishing common labels for their individual knowledge of inputs and outcomes. Such an agreement process includes several cycles of exchanging prototypes and/or single episodes. For instance an agent A provides a prototype description *d* to another agent B, B tries to match *d* to its own experience base. Given a successful match, the agents choose a common label depending on the existence of labels from previous negotiations.

In case the match at prototype level is not successful, the agents resort to instance level. This may lead to the creation of new prototypes in the agents which form a new basis for establishing a common label. If the negotiation at instance level fails, the agents' current experience bases are too distant. This however may change with further acquisition of knowledge. Using a set of labels, agents can exchange information more efficiently than by exchanging prototype representations or instance data. Since the shared labels are grounded in each agent's experience due to virtual embodiment and the affordance approach, knowledge exchange is possible even though the agents do not share the same internal representations.

With the agreed labels, agents can exchange task solving strategies: which tools or tool chains to use in a given situation, how to react on a given input, or reach a desired outcome. For example, an agent may present user input to another agent and ask for a recommendation what tool to use. This reduces the search space of individual agents for finding applicable tool or tool chain usage and increases the probability of satisfying the user.

With the presented affordance-inspired approach, we have established a basis which allows us to systematically explore how the design of an environment and the sensor channels of a virtual agent influence the acquisition of knowledge via self-experience, how grounded and negotiated labels can lead to shared knowledge among individual agents, how the distribution of knowledge in groups of agents is influenced by different strategies of label agreement, and which strategies of communication and learning best increase the agent's capabilities in assisting the user.

Acknowledgments

This research is supported by the EC Cognitive Systems Project IST-027596-2004 RASCALLI, and by the Federal Ministry of Economics and Labour of the Republic of Austria.

References

Irran J., Kintzler F., Pölz P., 2006, Grounding Affordances. In proceedings: Trappl R. (ed.): Cybernetics and Systems 2006, Vienna: Austrian Society for Cybernetic Studies, ISBN 3 85206 172 5

James J. Gibson, 1986, The ecological approach to visual perception. Hillsdale, New Jersey; London: Laurence Erlbaum Associates.

Kintzler, F. et al., 2007. Affordance-related Robotics Research – A Survey . Journal for research on adaptive behaviour in animals and autonomous, artificial systems (submitted).

Neural Associative Memories for Language Understanding and Action Planning in a Robotics Scenario

Heiner Markert[1], Ulrich Kaufmann[1], Zoehre Kara Kayikci[1] and Günther Palm[1]

[1]University of Ulm
Institure of Neural Information Processing
Oberer Eselsberg, D-89069 Ulm, Germany
Tel.: (+49)-731-50-24151; Fax.: (+49)-731-24156
heiner.markert@uni-ulm.de

Understanding language is a key feature required by many robotics and human assistance applications in order to simplify human-machine interaction. Furthermore, from a brain researchers perspective, language is one of the most exciting and fundamental features of the human brain. We have implemented a biologically inspired approach to language understanding utilizing Hidden Markov models and neural associative memories in a robotics scenario (Fay et al. 2005). The robot demonstrates its understanding of simple command sentences like “bot show plum” by performing corresponding actions. The scenario is a robot standing near to a white table with different objects lying on it. The robot receives spoken command sentences and has to determine their meaning and perform a corresponding action (see Figure 1).
The system uses a standard Hidden Markov Model (Rabiner and Juang 1993) approach for basic recognition of phonetic features (triphones) and forwards the extracted feature stream to a neural network system. Besides our general interest in brain research, there are also technical reasons that support the idea to implement a neural system: Neural networks are able to flexibly handle and interpret ambiguous or incomplete input and thus are a natural choice for systems that require a high amount of fault tolerance.

Figure 1: The robot is standing near to a table with objects on it. An operator gives spoken command sentences like "bot show plum". The robot has to demonstrate its understanding by performing a corresponding action.

The network in our system consists of several modules, each of which is implemented using binary associative memories (Palm 1982). The networks architecture can roughly be partitioned into the following three main parts:

1. word recognition network: translates the stream of triphones into word representations
2. sentence recognition network: validates the stream of words against a given grammar
3. action planning network: initiates actions required to fulfil a correctly understood command sentence

There is also a radial basis function based object recognition network (Kaufmann et al. 2005) in the system that is used by the action planning part. For an overview of the model architecture, see Figure 2. Note that the connections between the modules are mainly going in one direction and there is not much feedback used yet. However, each connection can forward ambiguous states, i.e. if the word recognition network could not decide for a unique interpretation of the phonetic input, it can forward several alternatives to the sentence recognition network which can then try to decide for one of the alternatives from additional contextual information.
The word recognition network extracts words from the stream of triphones received from the Hidden Markov model. The network uses various prediction mechanisms in order to improve error correction capabilities, for example, a list of all words matching the current input is used to predict and probably correct input in the next step.
The sentence recognition network validates the stream of words received from the word recognition part with respect to a given grammar. This part of the network can also resolve ambiguities on single word level with the help of contextual information. For example, the sentence “bot lift bwall” with an ambiguous input between “ball” and “wall” in the last word can be resolved to “bot lift ball” because a wall is not liftable. Similarly, ambiguities on the grammar level can be

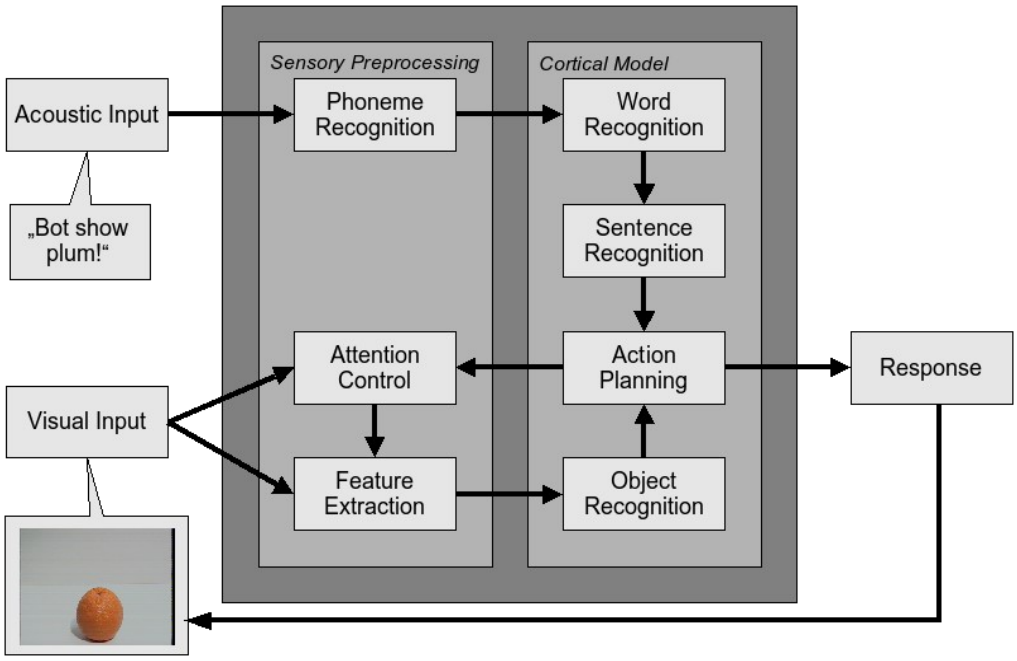


Figure 2: Overview of the system. Sensory preprocessing (left) is performed using standard computer science algorithms, “Cortical model” (right) is built using neural networks.

resolved: Consider the sentence “bot put orange orange orange plum” which is valid with respect to our simplified grammar that does not contain adpositions such as “to”. The sentence means that the robot should put an orange orange to an orange plum. However, until the word “plum” is heard and understood by the system, it is not clear which one of the “orange”-instances is going to be an object or an adjective. As long as a unique interpretation is possible, both types of ambiguities might also be combined and can still be resolved.

The action planning network coordinates simple motor programs in order to perform various actions. The robot can basically drive its wheels, search for objects and lift and drop them, which can be combined to actions like “go (to) table/object”, “show object”, “lift object”, “drop object” or “put object1 (to) object2”. The action planning network initiates the corresponding motor programs in order to perform the task at hand. It constantly evaluates the sensor inputs to determine when a subgoal is finished and another basic action has to be performed in order to reach the global goal given by the operator.

To our knowledge this is the first robot control architecture including simple language understanding, visual object recognition and action planning, that is realized completely by neural networks and that is able to resolve ambiguities and to learn new words (Markert et al. 2007). It also represents the first real-time functional simulation of populations of spiking neurons in more than seven cortex areas in cooperation.

References

Fay, R., Kaufmann, U., Knoblauch, A., Markert, H. and Palm, G. 2005. Combining Visual Attention, Object Recognition and Associative Information Processing in a NeuroBotic System. In Wermter, S., Palm, G., Elshaw, M. (Eds): Biomimetic Neural Learning for Intelligent Robots. Springer LNAI 3575

Rabiner, L. and Juang B.H. 1993. Fundamentals of speech recognition, Prentice-Hall

Palm, G. 1982. Neural Assemblies, Springer

Kaufmann, U., Fay, R., Markert, H. and Palm, G. 2005. Neural Networks for Visual Object Recognition Based on Selective Attention. In Proceedings of the International SenseMaker Workshop on Life-Like Perception Systems

Markert, H., Knoblauch, A. and Palm, G. 2007. Modelling of syntactical processing in the cortex. BioSystems 89, 300-315

Mapping the vocal tract with a 2D vocalic articulatory space: applications to developmental robotics

Cláudia Soares, Alexandre Bernardino *

Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal
csoares@isr.ist.utl.pt, alex@isr.ist.utl.pt

Abstract

Articulatory speech synthesis has been used recently to emulate in robots the speech production and learning capabilities of human infants. Acoustic to motor maps are created by babbling strategies, exploring the available motor degrees of freedom and creating associations to the listened sounds. However, the physiology of the human vocal tract contains many redundant parameters, which poses problems in sensor-motor map learning. In this paper we show that vocalic speech requires, in fact, a very reduced number of parameters and, based on linguistic knowledge, propose a two-dimensional articulatory space. The proposed space is generated through the convex combination of prototype vowels representing extremal points in the articulatory parameters. We show experimentally, using a known articulatory synthesizer, that the proposed model production space is enough to generate most of the vowel acoustic subspace, in terms of the Mel Cepstral Coefficients' variance. This provides a low-dimensional and intuitive vowel production space, suited for automatic production, recognition and learning of speech in articulatory models.

Introduction

Developmental robotics aims at studying how knowledge on human cognitive development can be exploited to allow robot to learn and adapt continuously to its morphology and environment (Lungarella et al., 2003). The development of speech production involves the exploration of the vocal tract capabilities during the infants early developmental stages. Also for speech perception development, the vocal tracts articulatory information may be of fundamental importance. The Motor Theory of Speech Perception (Liberman and Mattingly, 1985) supports that the basic units of speech perception are the intended phonetic gestures of the speaker, represented in the brain as invariant motor commands that call for movements of the articulator. According to this theory, speech would be perceived by inferring the articulatory shape of the vocal tract from the acoustic signal, and performing recognition in the motor space. The rationale for this approach comes from the fact that motor commands, on the contrary of acoustic signals, are invariant to the environmental conditions, thus providing stable references for recognition.

*This work was supported by EU NEST Project 5010 - Contact, and by Fundação para a Ciência e a Tecnologia (ISR/IST plurianual funding) through the POS_Conhecimento Program that includes FEDER funds.

Initial experimental evidence for the importance of motor information in recognition tasks started with neurophysiological recordings in neurons of the pre-motor cortex of primates, which led to the discovery of Mirror neurons (Gallese et al., 1996). These neurons show spiking activity both when the monkey executes and observes a grasping movement. An experimental study, with a robotic artifact, for the recognition of grasping gestures (Lopes and Santos-Victor, 2005), showed drastic improvements when recognition was based on the motor space rather than the visual space. Mirror neurons are located in the ventral premotor cortex, possibly the homologue of Broca's area in humans, which led to the speculation that action recognition and language production share a common system. Neuroimaging studies of the Broca's region have recently supported this hypothesis in a joint action recognition, language production and grasping task (Hamzei et al., 2003).

Such a theory represents a novel paradigm for speech perception but poses novel challenges since it requires the availability of the agents' motor signals and learning mechanisms for associating the motor and auditory spaces. This can be achieved by exploratory learning (spanning the agent's motor space and observing the outcome in auditory terms), or by imitative learning (listening to other agent's produced sounds and trying to imitate). But, depending on the dimensionality on the involved spaces, this may be too complex to do in practice. A recent model for the control of speech production in humans, the Diva Model (Guenther et al., 2006), follows the motor theory paradigm and accounts for a wide range of acoustic, kinematic and neuroimaging human data. Sensory-motor association is done locally by computing the tangent spaces to the synthesis function at some prototypical points. Mapping the whole articulatory space would require a lot of exploratory learning which, in high dimension spaces, becomes impractical.

In this paper we propose a methodology to create an articulatory subspace in vowel production, allowing a complete characterization of the speech synthesis function and its properties, permitting an feasible online speech processing and learning for robots as Chico and Chica (depicted in Figure 1). The method is motivated by results of Linguistics and Phonetics, where the vowel space is represented in motor terms in a 2D representation. We

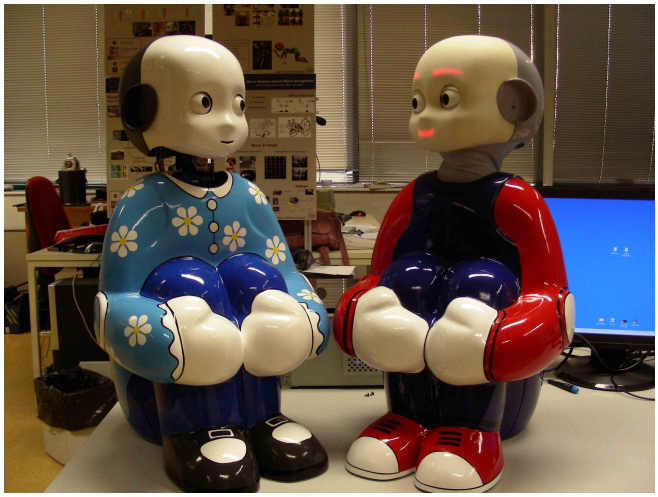


Figure 1: Humanoid robots must interact with each other and with humans by spoken language. These are the robotic platforms for the implementation of the algorithm. Some work has been already done with these robots in speech perception (*vide* (Hörnstein and Santos-Victor, 2007))

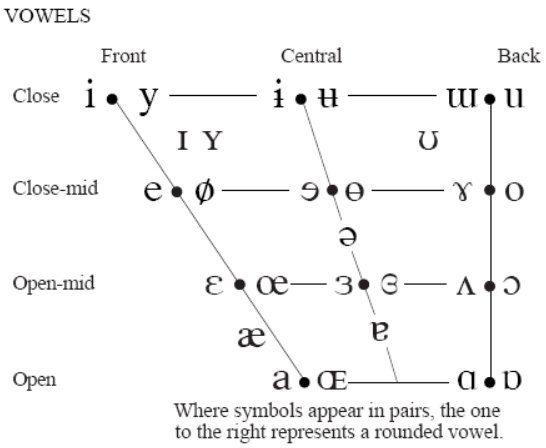
show that a 2-dimensional plane generated by the convex combination of 3 extremal motor primitives is able to adequately represent the vowel acoustic space. An additional advantage is that, since the synthesis is based on two sole articulatory parameters, it is easy and intuitive to graphically visualize the motor-to-acoustic manifold, allowing a better characterization of its properties.

The paper is organized as follows. Section *Linguistic Motivation* briefly presents the linguistics and phonetics results motivating our approach. Then, in Section *The Speech Production Model* we describe the articulatory speech synthesizer used in this work and mathematically formulate the proposed articulatory dimensionality reduction principle. We have performed several experiments illustrating the validity of the approach, presented in Section *Experimental Results*. Finally, Section *Conclusions* present some conclusions and directions for future research.

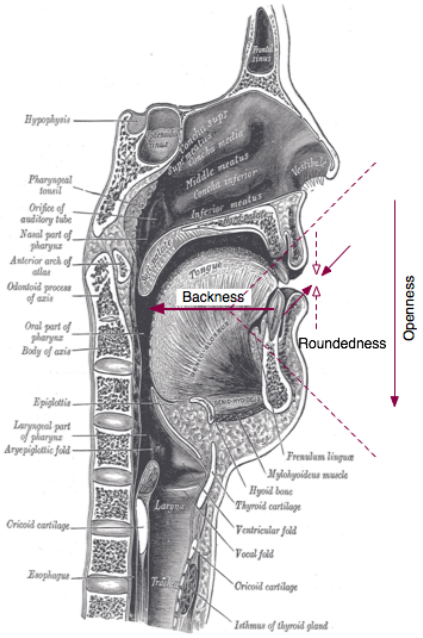
Linguistic Motivation

Since the beginning of Linguistics and Phonetics speech sounds are classified mainly by articulatory parameters. One of the pioneer works in defining where are vowels located in the articulatory space was (Jones, 1917) in which the mathematician and phonetician Daniel Jones first proposed the Cardinal Vowel Diagram. This diagram was subject of discussion and contributions from the phonetics community and gave rise to the unanimously accepted representation for oral vowels today.

The schematic in the International Phonetic Alphabet (IPA) for oral vowels in Figure 2(a) shows the distribution of vocalic sounds in three dimensions relative to the human vocal tract: height (vertical axis), backness (horizontal axis) and roundedness (lip rounding)(Association,



(a) International Phonetic Alphabet chart for oral vowels.



(b) Main degrees of freedom represented in the IPA chart. Figure from (Gray, 1918), with our labels.

Figure 2: Articulatory degrees of freedom in the IPA chart representation.

1999) as illustrated in Figure 2(b).

This choice of reference frame has roots in the physiology of the phonatory system. The vocal tract configuration for oral vowels is function of the tongue, the jaw and the lips. The jaw and lips can have several degrees of openness, the tongue can assume the articulatory positions in front, center or back of the oral cavity and the lips can also change the vocal tract by rounding. So, these three articulatory parameters are considered the main degrees of freedom of vocalic speech sounds, and represent the directions that better explain the inter-vowel variation. Nevertheless, there are other static articulatory parameters that influence oral vowel quality, although they are not determinant in most spoken languages.

Proc. LangRo'2007
978-972-96895-2-9

In most languages, rounded and unrounded vowels are not minimal pairs, i.e., for the same articulatory configuration, roundedness alone does not create two different phonological vowels. In addition to this, some studies support that roundedness is perceived mainly by vision in normal hearing-seeing subjects (Traunmüller, 2006), For these reasons, the main articulatory dimensions considered for oral vocalic sounds in the human vocal tract are the height and backness, motivating the approximation proposed in this paper — whatever the dimensionality of the articulatory space we consider, there is an two-dimensional subspace approximation that maps the vowel system of most languages. The phones [i], [a] and [u] define a set of axis in the 2D plane of the articulatory parameters of *height* and *backness*. These three vowels are called *corner vowels* because they represent extreme placements of the tongue forming the corners of a triangle in articulatory space. They also form a triangle in formant space (F1 – F2)(Titze, 1994). Therefore, we consider these phones the extremal points in our model, and will produce the remaining ones by their convex combination. This will be detailed in the following Section.

The Speech Production Model

To test and validate our proposal we use a well-known articulatory speech synthesizer. This will allow us to do systematic tests and quantify the errors arising from the proposed approximation. From realizations of the extremal phones [i], [a] and [u], we generate a dense representation of the feasible acoustic signals. Then, to evaluate the model, we compute the acoustic errors outside the feasible set.

Articulatory Synthesizer

The synthesizer in use[1] is a Matlab version of Shinji Maeda's Vocal Tract Calculator (*VTCalcs*) (Maeda, 1990). The seven articulatory parameters are *jaw, tongue, shape, apex, lip_ht* (lip hight), *lip_pr*, (lip protrusion), *larynx*. Each one can assume any value in $[-3;3]$. The articulator parameters are presumed independent, which is not the case in the human vocal tract, leading sometimes to improbable configurations of the articulators, producing a non human sound or even no sound at all. In fact, after a dense sampling of the six-dimensional hypercube and feeding the samples to the synthesizer, as explained later in this section, we realized that only 44.22% of the articulatory vectors generated sound, even if not a human-like one.

The space of the articulators in *VTCalcs* is homographic to $\mathbb{R}^7$, but to produce vocalic voiced sounds only 6 parameters are distinctive, since larynx controls the voicing.

The synthesizer's output is a sound represented by its temporal amplitude. To analyze the sound waveform we use the Mel Frequency Cepstral Coefficients (MFCC) (Davis and Mermelstein, 1980), using 12 coefficients.

[1] Available at the CNS Speech Lab webpage `http://speechlab.bu.edu/VTCalcs.php`

Let vector $\mathbf{v} \in \mathcal{V} \subset \mathbb{R}^6$ represent a configuration of the six-dimensional synthesizer's articulatory space and $\mathbf{a} \in \mathcal{A} \subset \mathbb{R}^{12}$ be a vector of MFCC coefficients in the acoustic space. We define the synthesis function as:

$$f : \mathcal{V} \mapsto \mathcal{A}, \quad \mathbf{a} = f(\mathbf{v}) \tag{1}$$

The function is not invertible — distinct articulatory configurations may lead to very similar sounds (in particular, many configurations generate no sound at all). Therefore, there is ambiguity in the identification of motor configurations corresponding to the listened acoustic signals, which may pose problems to motor-based learning and recognition algorithms. To deal with this we define a subspace of $\mathcal{V}$ where the restriction of f to this subspace is assumed invertible.

Dimensionality Reduction

We define a two-dimensional subspace of the full articulatory space, generated by a convex combination of vowels corresponding to extremal positions in the articulatory space. There are two major arguments that support this approach: a linguistic argument, and a experimental one. As mentioned in Section *Linguistic Motivation*, according to Linguistics and Phonetics knowledge, most of the vowel production capabilities of the human vocal tract can be explained by two parameters related to the height and backness of the articulators. The experimental argument is that the Isomap, as discussed in Section *Experimental Results*, shows that there is a good two dimensional approximation to the image of f.

Considering the $\mathbb{R}^6$ prototypes for the extremal phones [i],[a] and [u], it is possible to generate an affine space with all the properties of a convex space. Let a_0, u_0 and $i_0 \in \mathbb{R}^6$ be the chosen vowel prototypes for [i], [u] and [a] and a two-dimensional vector $\mathbf{p} \in \mathcal{V} : \mathbf{p} = (\alpha, \beta)$, with α and β real parameters. A convex combination of the given points forming a 2-dimensional triangle, can be defined by the function:

$$v : \mathcal{P} \subset \mathbb{R}^2 \mapsto \mathcal{M} \subset \mathcal{V}$$
$$v(\alpha, \beta) = \alpha\, i_0 + \beta\, a_0 + (1 - \alpha - \beta)\, u_0$$

where the input space $\mathcal{P}$ is defined as:

$$\mathcal{P} = \{(\alpha, \beta) : \alpha + \beta \leq 1 \wedge \alpha, \beta \geq 0\}$$

Let $\mathcal{M}$ be the image of v, and denote it the *Motor Space*. We define the function f_2 as the restriction of the synthesizer's function f to the motor space, and call it's image $\mathcal{A}_2$

$$f_2 : \mathcal{M} \mapsto \mathcal{A}_2 \subset \mathcal{A}. \tag{2}$$

We will denote f_2 as the *Motor-Acoustic Map*. The image of this function will produce a 2D manifold $\mathcal{A}_2$ in the MFCC acoustic space. Given the choice of the *Motor-Space*, the properties of the used synthesizer (assuming smoothness), and the dense sampling made on $\mathcal{M}$, there are strong reasons to believe that f_2 is invertible. Therefore, the inverse function of f_2, f_2^{-1} is an acoustic to motor

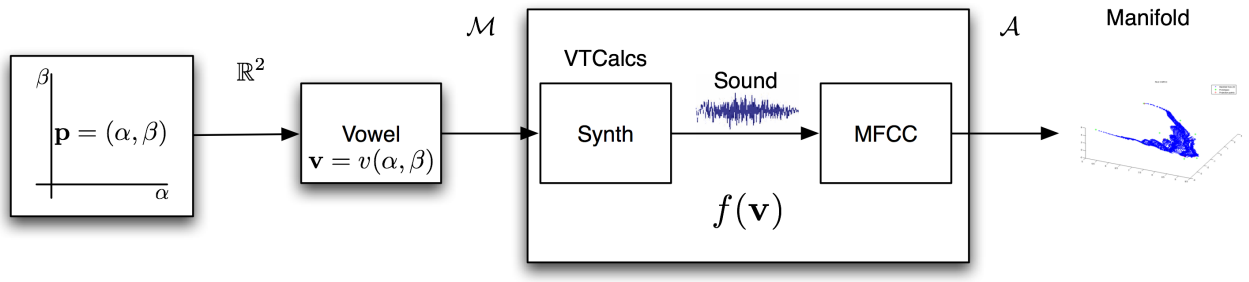


Figure 3: Vowel generation diagram.

map. A schematic representation of the proposed vowel production model is shown in Figure 3.

The twelve-dimensional acoustic space was sampled twice; one representing the span of the reduced articulatory space (using the motor map f_2 from the motor space $\mathcal{M}$), and another representing the span of the full articulatory space (from $\mathcal{V}$). We will show that the former contains most of the information present in the latter.

To estimate the *acoustic manifold* $\mathcal{A}_2$ we have sampled the parameter space $\mathcal{P}$ in steps of 0.01 in the α and β parameters, generating a discrete set of 5000 samples:

$$\mathcal{P}_d = \{\mathbf{p}_i,\ i = 1,\ldots,5000\}$$

These samples were then used to generate a motor-space sample set, using function v:

$$\mathcal{M}_d = \{\mathbf{m}_i = v(\mathbf{m}_i),\ i = 1,\ldots,5000\}$$

Thus a discrete sampling of the acoustic manifold was created using the synthesizer's function:

$$\mathcal{A}_{2d} = \{\mathbf{a}_i = f_2(\mathbf{m}_i),\ i = 1,\ldots,5000\} \qquad (3)$$

The first three coordinates of the sampled acoustic manifold are plotted in Figure 4.

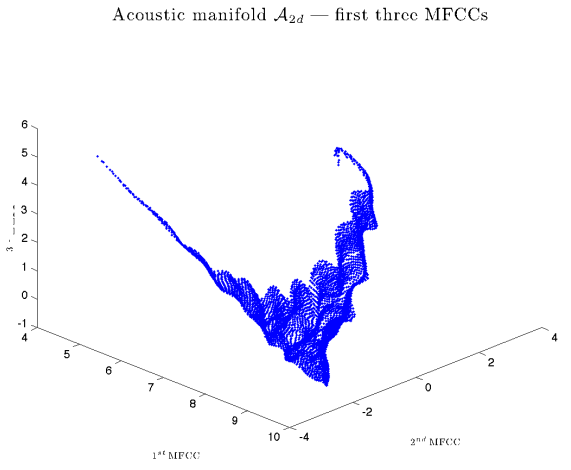


Figure 4: Representation of the first three Mel coefficients of the acoustic manifold.

The *VTCalcs* parameter's six-dimensional $\mathcal{V}$ space was also sampled in steps of 0.6 obtaining a grid with 10 samples per dimension. The point cloud has 10^6 samples:

$$\mathcal{V}_d = \left\{\mathbf{v}_i,\ i = 1,\ldots,10^6\right\}$$

Again, the synthesizer's function was applied to the data;

$$\mathcal{A}_d = \left\{\mathbf{a}_i = f(\mathbf{v}_i),\ i = 1,\ldots,10^6\right\} \qquad (4)$$

From this data it was removed the set of samples with zero sound amplitude, retaining 44.22% of the initial number.

Experimental Results

To validate the proposed model we generate a set of test vowels $\mathbf{a}^t$ and compute the error in acoustic space (MFCC coefficients) between each one and its projection on the manifold $\mathcal{A}_{2d}$. We also consider the residual variance incurred in a two dimensional approximation of $\mathcal{A}$.

Since we do not have a analytic expression for the $\mathcal{A}_2$ surface, we use its sampled version defined by equation (3). To compute the projection of each point we use the nearest neighbor operator:

$$nn(\mathbf{a}^t) = \left\{\mathbf{a}_i \in \mathcal{A}_{2d} : i = \underset{i}{\operatorname{argmin}} \left\{\|\mathbf{a}_i - \mathbf{a}^t\|_2\right\}\right\} \qquad (5)$$

The acoustic approximation error is then computed by:

$$E_a(\mathbf{a}^t) = \|\mathbf{a}^t - nn(\mathbf{a}^t)\|_2 \qquad (6)$$

The acoustic approximation error relative to the size of the manifold is defined as

$$\delta_a(\mathbf{a}^t) = \frac{E_a(\mathbf{a}^t)}{\max(length(\mathcal{A}_{2d}))}\ 100\% \qquad (7)$$

This measure is dimensionless and gives an indication of how good is the approximation relative to the size of the approximating surface. We consider acceptable to use the maximum length of $\mathcal{A}_{2d}$ to normalize the error because the manifold's shape is not too discrepant, as it is possible to confirm in the Isomap embedding shown in Figure 5. This embedding was determined with the Isomap algorithm

Proc. LangRo'2007
978-972-96895-2-9

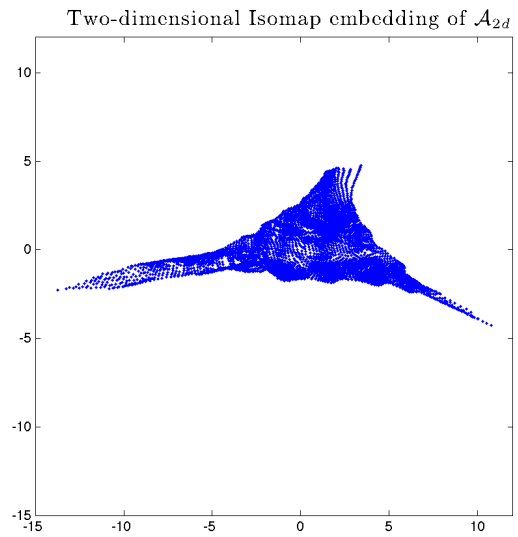


Figure 5: Isomap embedding for the two-dimensional manifold $\mathcal{A}_{2d}$.

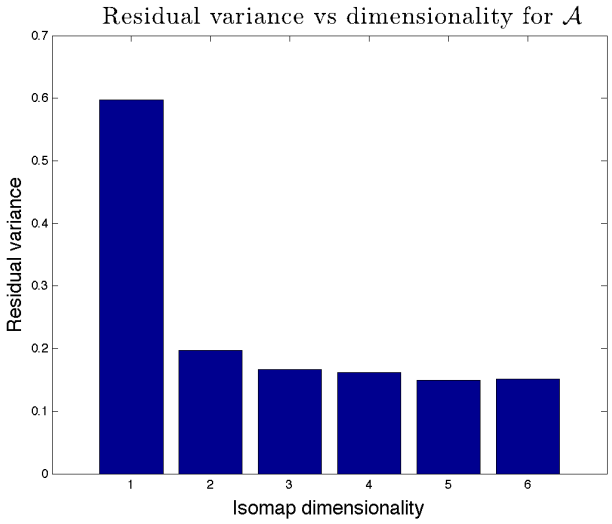


Figure 6: The Isomap algorithm provides the residual variance of the fit to the model's dimensionality. The greatest decrease in variance happens from one to two dimensions of the manifold representing the global acoustic space $\mathcal{A}$.

as described in (Tenenbaum et al., 2000). The *isometric feature mapping procedure* or Isomap recovers low-dimensional nonlinear structure in perceptual datasets. It finds a space embedding for the data, preserving its intrinsic metrics, by conserving distances measured through *geodesic paths* along the observation manifold. For $\mathcal{A}_{2d}$, Isomap created a reproduction, in the two-dimensional space, of the pairwise distances measured in the acoustic twelve-dimensional space.

Dimensionality reduction: validation

To validate the goodness of a two-dimensional approximation for the full space $\mathcal{A}$, the dimensionality of the sampled space $\mathcal{A}_d$, defined in equation (4), was investigated.

Through Isomap we estimate that the dimensionality of the image of f is 2, with a residual variance of 0.197, as illustrated in Figure 6.

The global articulatory space $\mathcal{M}$ is six-dimensional, thus the maximum possible dimensionality for $\mathcal{A}$ is six because f is continuous. The residual variance of the data for six or more dimensions can be interpreted with regard to phenomena such as noise and numerical problems in the MFCCs calculation.

This experimental result confirms that there is a good two dimensional approximation to the overall acoustic space $\mathcal{A}$. The residual variance present in the 2D approximation is partially due to the model simplification but its slow decrease with dimensionality leads to the conclusion that it is caused mainly by non informative phenomena.

Vowel prototypes: appropriateness

To investigate the performance of the approximating space with speech sounds of real languages, some experiments have been conducted with synthesized prototypes of several languages. Those prototypes may lie outside the motor space $\mathcal{M}$ because there are many redundant articulatory configurations that generate the same vocalic sound. We want to show here that $\mathcal{M}$ is complete, i.e. it contains a configuration generating an (almost) identical sound.

Some prototype vowels used in the tests are included in the *VTCalcs* matlab package and are preexistent to the experiment; the other sets were constructed by us and validated by naive native speakers. The speech sounds intensity, fundamental frequency and duration were kept constant so to validate strictly the model for vocal tract configuration.

In the *VTCalcs* package there are eleven prototypes for oral vowels which are found outside the two-dimensional polygon $\mathcal{M}$. They were used to evaluate the amount of error introduced in the two-dimensional approximation. The error was measured as described above, and the results are shown in Table 1. The oral vowels from two very distinct european languages were also used for the same purpose: vowels from Portuguese, an indo-european, romanic language, and vowels from Finnish, a finno-ungric language. Nine Portuguese prototype vowels were used. The errors are shown in Table 2. From Finnish, the eight short vowels were investigated, with results that can be seen in Table 3.

The sample mean over the percent error $\delta_a(\mathbf{a}^t)$ is 2.95% in the portuguese vowels set, 3.87% in the finnish vowels, and 2.23% in the *VTCalcs* set. The standard deviation is 2.22%, 2.81% and 2.02% in the portuguese, finnish and *VTCalcs* sets, respectively. The maximum value for the percent error is 9.17% in the finnish dataset.

So, in terms of the error, the two-dimensional convex space performs well with linguistically relevant synthesized speech sounds. Acoustically, the prototypes and the projections are hardly distinguishable. Inverting the projected points through f_2^{-1} back to the two dimensional motor space $\mathcal{M}$, and plotting the result (Figure 7) makes it

Table 1: *Approximation error for the* VTCalcs *prototypes.*

vowel	symbol	$E_a(\mathbf{a}^t)$	$\delta_a(\mathbf{a}^t)\%$
1	iy	0.40149	1.6295
2	ey	0.17829	0.72361
3	eh	0.1522	0.61771
4	ah	0.48633	1.9738
5	aa	0.24348	0.98818
6	ao	0.51035	2.0713
7	oh	0.58974	2.3935
8	uw	1.6111	6.5389
9	iw	1.4057	5.7053
10	ew	0.29547	1.1992
11	oe	0.18119	0.73536

Table 2: *Approximation error for the portuguese prototypes.*

vowel	IPA symbol	$E_a(\mathbf{a}^t)$	$\delta_a(\mathbf{a}^t)\%$
1	i	0.13425	0.54487
2	ɐ	1.2335	5.0061
3	ɛ	0.37961	1.5406
4	ɔ	0.50396	2.0453
5	e	0.61689	2.5037
6	o	1.4141	5.739
7	a	0.24161	0.98057
8	u	1.6211	6.5792
9	i	0.39633	1.6085

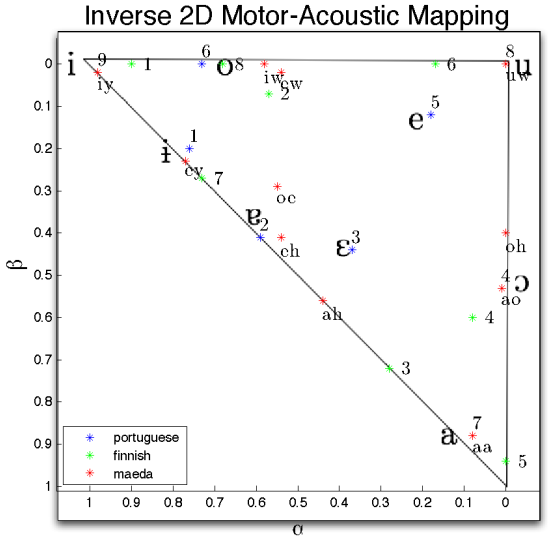


Figure 7: The inverse mapping of the vowel prototypes. The Portuguese vowels are numbered as in Table 2, and the Finnish as in Table 3. Some landmark IPA phonetic symbols are also represented.

is possible to extract some similarities between the IPA openness and backness and the motor space α and β parameters. The hypothesis that the restrictions in the construction of $\mathcal{M}$ can be used to simulate physiological constraints, is corroborated by these experimental results.

Table 3: *Approximation error for the finnish prototypes.*

vowel	IPA symbol	$E_a(\mathbf{a}^t)$	$\delta_a(\mathbf{a}^t)\%$
1	i	0.28764	1.1674
2	ø̞	0.7918	3.2135
3	æ	0.99949	4.0564
4	ɑ̞	0.87593	3.555
5	ɑ	1.6373	6.645
6	u	0.5645	2.291
7	e̞	0.21044	0.85406
8	y	2.2605	9.1741

Conclusions

In this paper we have proposed a two dimensional parameterization for the motor space of an available speech synthesizer, *VTCalcs*. The approach is able to generate acoustic signals that represent well all the vowels produced by the synthesizer. Namely, the euclidean error relative to the size of the two dimensional approximating surface has an average of about 3% and a maximum of 9.17% in the used test sets, and the Isomap analysis of the residual variance versus the dimensionality of the approximating manifold confirms the validity of a two-dimensional model for the overall acoustic space.

The proposed model is important by two main reasons:

- The motor space is two-dimensional, thus can be densely sampled with low computational requirements. This simplifies creation and representation of the motor acoustic map.
- The restriction of the synthesizer's function to the proposed motor-space is invertible, allowing to map signals back from the acoustic to motor coordinates.

In future work we will apply the proposed model in the early stages of autonomous speech learning of humanoid robots. The fact that this space has low dimensionality facilitates initial bootstrapping. We will also consider the problem of Mel Coefficients robustness and normalization procedures on the signals.

Since the acoustic manifold appears to be smooth, we will provide it with a differential structure and use it for local optimization, e.g. for guided exploratory learning in imitation tasks. In the long term we intend to apply the proposed model in the early stages of autonomous speech learning of an humanoid robot. The fact that this space has a dimensionality of two, facilitates its bootstrapping role in autonomously to produce and recognize speech. Once the system learns a good initial model of the motor-audio map using the low dimensional manifold, it can expand the available degrees of freedom and refine its production capabilities. As in the ontogenesis of humans infants, such a developmental strategy is more likely to succeed than learning from scratch with the whole system's complexity.

Proc. LangRo'2007
978-972-96895-2-9

References

Association, I. P. (1999). *Handbook of the International Phonetic Association*. CUP, Cambridge.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 28(4):357–366.

Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119(2):593–609.

Gray, H. (1918). *Anatomy of the Human Body*. Lea & Febiger, Philadelphia.

Guenther, F. H., Ghosh, S. S., and Tourville, J. A. (2006). Neural modeling and imaging of the cortical interactions underlying syllable production. *Brain and Language*, 96(3):280–301.

Hamzei, F., Rijntjes, M., Dettmers, C., Glauche, V., Weiller, C., and Buchel, C. (2003). The human action recognition system and its relationship to broca's area: an fmri study. *NeuroImage*, 19(3):637–644.

Hörnstein, J. and Santos-Victor, J. (2007). A unified approach to speech production and recognition based on articulatory motor representations. IROS07, to appear.

Jones, D. (1917). An english pronouncing dictionary. In *Daniel Jones: Selected Works*. Routledge, London.

Liberman, A. M. and Mattingly, I. G. (1985). The motor theory of speech perception revised. *Cognition*, 21(1):1–36.

Lopes, M. and Santos-Victor, J. (2005). Visual learning by imitation with motor representations. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 35(3):438–449.

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15(4):151 — 190.

Maeda, S. (1990). *Speech production and speech modeling*, chapter Compensatory articulation during speech, evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model, pages 131 — 149. NATO ASI Series. Kluwer Academic Publisher, Dordrecht, Netherlands.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Titze, I. R. (1994). *Principles of Voice Production*. Prentice Hall, Eaglewood Cliffs, New Jersey.

Traunmüller, H. (2006). Cross-modal interactions in visual as opposed to auditory perception of vowels. In *Proceedings of Fonetik 2006, the XIXth Swedish Phonetics Conference*, pages 137—140. Department of Linguistics, Lund University.

Multi-Robot Interaction using Finite State Grammars *

João Xavier and Urbano Nunes
Institute of Systems and Robotics - ISR, University of Coimbra, Portugal
smogzer@gmail.com, urbano@isr.uc.pt

Abstract

In this work is introduced a set of software components for Human Robot Interaction (HRI) based on Finite State Grammars (FSG), like Speech Recognition, keyboard input or even task scripts. Normally FSG inputs like speech recognition based interfaces interact only with navigation devices. In this work is presented a set of components that allows the developer to easily extend this interaction to any device. Task Contexts and Awareness (both spatial and situational) are provided so that the system is reactive to events in the perceived world. Components then use this common information structure to perform actions on the surrounding interactive environment. The synergy obtained from different system components can be used for interacting also with key assistive technologies, such as domotics, emergency management, driver support aids, etc. Is demonstrated a scenario with a user interacting with multiple robots that demonstrates some of the concepts of the article. The software is available on-line with a LGPL license so that others can benefit and contribute to it as well.

Introduction

The work presented herein started as a way to increase the interaction possibilities of the Robchair [Pires and Nunes, 2002], autonomous robotic wheelchair. Speech based interaction can be one of the few possibilities people with physical disabilities have to interact with their wheelchairs and the appliances they use. Speech based interaction has the advantage of leaving the hands of the user free for other tasks, and because the commands available via voice are broader, the interaction has more possibilities than the limited degrees of freedom of a joystick.

A finite state grammar (FSG) is one that maps language to a state diagram which starts in an initial state, and each step produces a word (or element; phoneme, syllable, etc.) with each transition. At the final state, the sequence of words produced is a sentence or phrase. In a complete finite stage grammar, we can theoretically produce any sentence by tracing a path from left to right, from the initial state to the final state. Transitions can also be closed loops allowing for an infinite number sentences.

Normally FSG based interfaces, that can be speech recognition ones, interact only with the navigation of the robot. In this work is presented a FSG based interface that dynamically adapts to the robot environment, where the lexicon of the commands reflects the abilities of interaction with the surrounding environment, and that is not limited to control a single robot.

A light-weight cognitive layer is presented in the form of specialized components that provide services like spatial and situational awareness and also cognitive and topological descriptions of the environment. This cognitive layer components allow a common user to ask the robots to execute tasks, without caring for the encapsulated low-level algorithms that the component will use. It is possible to define tasks and behaviors that can apply to many robots just by modifying XML configuration files.

In the lab it is very practical for a single user to interact with a large number of devices e.g. managing a robotic soccer team; it also proves practical in experiments that require the user to stay far from the computer/robot e.g. during camera calibration while the user is holding a chessboard he can instruct the computer to capture images while he moves the chessboard.

The implementation is made as a set of components for the Experimental Robotics Framework (ERF) [Xavier, 2007]. The developed software integrates with the Player framework [Gerkey et al., 2003] for accessing real robots and also simulated ones in the simulators Player provides. For speech recognition is used the library CMU Sphinx [CMU, 2001].

In this article some words are written in CAPS and others in `teletype`. Capitalized words represent FSG words, and teletyped words represent C++ classes.

Paper structure

This article is organized as follows. Section presents the related work. Section describes the software architecture of the components. In Section a experiment is performed demonstrating the concepts used in the article. Future improvements are suggested in Section . Final remarks are given in Section .

Related work

This section describes the bibliography of research made in the field of speech based interaction, multi-robot interaction and spatial cognition and compares those works to

*This work was partialy supported by FCT (Portuguese Foundation for Science and Technology) under Grant:POSC/EEA-SRI/58279/2004.i

ours. In some bibliography it is possible to implement or wrap their methods into our system.

Multi-modality, Speech interaction

The lexicon used in various speech based interfaces for robotic wheelchairs is present in [Tellex and Roy, 2006]. A lexicon template used to control a single mobile robot is specified in [Zelek, 1997]. In that article the commands are formalized as specifying a verb, destination, direction, and speed. Our approach does not enforce a template for controlling the component or robots. Thanks to the FSG approach our lexicon is customizable and extensible. To allow this freedom we only specify the design of the task context, and the context selector that are the base for the multi-component system, and even these can be subclassed and still use the other components.

In [Flippo, 2003] is presented a multi-modal interface that was built to control mobile robots for military missions using spoken language, keyboard, mouse, touch, and gaze inputs, which can be used simultaneously. Although we have not provided touch or gaze inputs, besides the FSG methods we also have the possibility of interacting with the environment using the GUI.

Multi-robot interfacing

The task of interfacing with swarms of robots is discussed in [Halme et al., 1995] where the author introduces the concept of robot society. Both [Clark and Frew, 2003, Jones and Hinds, 2002] have developed 3D HRI for interfacing of a single user to multiple robots with support for motion planning and robot trajectory generation for target tracking. In [Makarenko et al., 2003] are studied the conditions under which the collaborative human involvement in shared HRI will not jeopardize scalability of the network of robots.

The multi-agent system (MAS) infrastructure, that combines HRI with a simulation environment for Human Search and Rescue (HSR) operations is described in [Nourbakhsh et al., 2005].

Context and Awareness

Context Acquisition is the mechanism used to understand the robots situation. The concept was first described in [Fong et al., 2004] but similar functionality can be identified in [Graves, 1995, Scholtz et al., 2004] that introduces the concept of Situational Awareness.

In [Chong et al., 2000] is described a system that uses vision to gain further spatial knowledge. In a previous project [Xavier et al., 2005] we perceived the robot surroundings, including persons, using data from a laser sensor.

Qualitative and quantitative representations of space are presented in [Kuipers, 2000]. Our approach for representations of the spatial knowledge is only based on tags, which are only qualitative. Although components can modify the tags of objects based on any classification algorithms they implement.

Some systems like Navchair [Levine et al., 1999] have different navigation modes depending of the situation. These modes are general obstacle avoidance, door passage, and automatic wall following, and are selected with a Bayesian network to decide what mode to activate based on the chair location.

That approach is possible in our system by having the spatial awareness load navigation components into ERF depending of the tags of the zone the robot is in. Examples of this is the docking/parking behavior that is available when the user is in the “ROOM“ zone and issues a ”DOCK“ command. The robot then switches the navigation algorithm to one that can plan complex maneuvers of the robot, like the ones based on Rapidly-exploring Random Trees [LaValle and Ku, 1999].

A robotic wheelchair with a speech interface that can understand the user intention in speech using the environmental information obtained from the range sensors is presented in [T. Iwase and Kuno, 2004]. Even if the user does not say details and issues a simple voice command, the wheelchair takes an appropriate action that the user expects. In our work we provide a task context where all the possible tasks of the robot can be issued according to the robot environment. This task context can be modified by the perception components like the one that detects persons presented in [Xavier et al., 2005].

In project Coyote [Skubic et al., 2004] is presented a mobile robot that can understand spatial commands and generate spatial linguistic descriptions of its environment. It can obey commands like “go to the right of the object” and describe where objects are relative to itself. In our system the spatial descriptions are given apriori by tagging the zones of interest, even if the zone is just the “outside” of another zone. The direction relative to world objects can be implemented with a combination of querying the position of the object and creating a new zone with tags corresponding to the relative position of the object.

Adjustable Autonomy [Birk and Pfingsthorn, 2006] defines layers of autonomy for interaction with the robots, where the higher layers mean more decisions by the robot, e.g. in the bottom layer the user modifies the robot velocities, while in the above layer the user just gives a target destination, and so on. This is equivalent in our system to the user opting to interface with a high-level task component or directly with a low-level robot interaction component.

System description

The system components can be organized in the following layers as shown in Fig. 1 : a cognitive layer that all robots have access to; a layer to organize the FSG interaction; a layer of robot tasks and behaviors; the world (either real or simulated) and the Player proxies that represent the sensors and actuators available for each robot.

The cognitive layer common to all the world agents, is where the perception of the world is gathered. This layer is sub-divided in two sub-layers, that are the event layer and the descriptive layer. The event layer contains com-

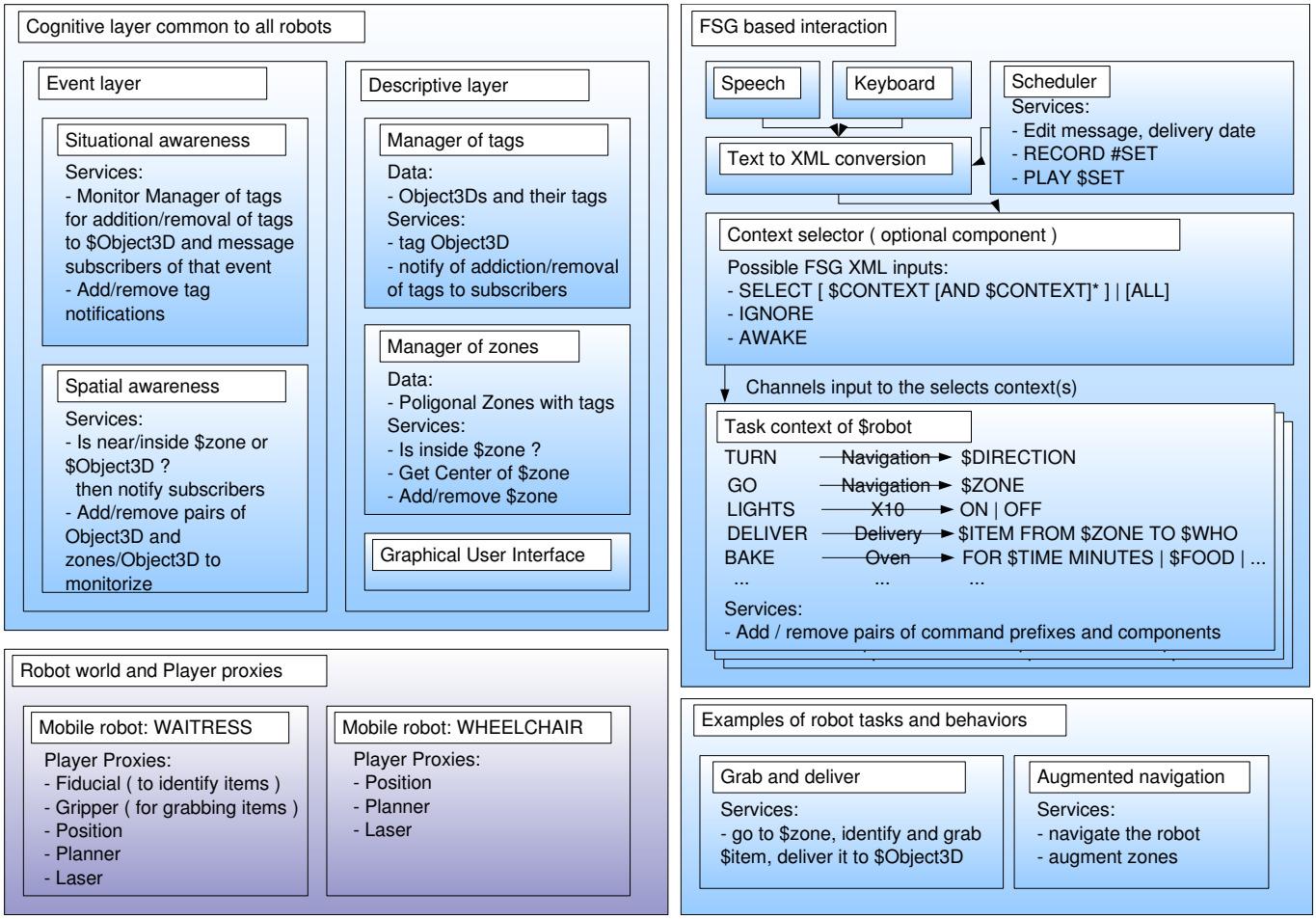


Figure 1: Overview of the framework components. A cognitive layer that is common to all the robots on the top left, on the top right the components that handle FSG inputs, on the bottom right are examples of robot tasks and on the bottom left is the robot world, along with the robots sensors and actuators.

ponents that monitor the state of the world and notify subscribers of changes. The descriptive layer as the name implies is used for describing world entities, it describes the geometry of polygon delimited zones and the tags of the `Object3D` entities. These tags descriptions can edited using the GUI.

The task and behaviors serve to encapsulate low-level algorithms so that these tasks can be used in batch commands for many robots or just for commodity. For the execution of tasks the robot access information from both the robot world, the cognitive layer components and can communicate with other agents/components.

The robot world layer contains Player/Stage proxies, such as the map, laser, sonar, position, planner, fiducial, etc.

Follows an introduction to ERF, and after a more detailed description of the components inside the main layers.

Introduction to ERF

To glue this layers together ERF is used. ERF provides basic support for all components can use and is extended through a component system that is modular, i.e. components can be (un)loaded at any time.

Components can query each another for state information or execute methods, and also of serialization so that they can save their states. In order to remove code redundancies, the components are specialized in certain tasks, e.g. the manager of zones that comprises all the methods relative to world zones. The queries are made exchanging XML formatted data. When the query reaches a target component, that component produces an answer, also in XML, and a report reply that informs if the question was acknowledged or if any error occurred.

The ERF library also contains OpenGL functions for drawing the GUI, and use the GUI for interaction with the world entities and robots.

Cognitive layer

This layer serves to map most environment information that can be used in robot tasks. It has two main blocks, one for reporting changes in the world, and the other to describe the world itself; both are described next.

Manager of tags Every object in the robot world derives from the class `Object3D`, which have a member `Tags`

Proc. LangRo'2007
978-972-96895-2-9

for distinguishing between different `Object3D` entities. `Tags` are implemented as a list of strings to which can be applied set theory.

Tags are added either directly by the user using the GUI - as supervisioned learning - or by scene interpretation components that make use of sensor data to create an `Object3D` in the robot world and tag it accordingly to what the algorithm perceives.

Some examples of tagging are marking a polygonal delimited zone with "LOUNGE", or "STAIRS", a moving blob with "PERSON" or something with "TARGET".

Components of the framework make use of the tags to interact with the world. Examples of interaction of component with the tags are:

- the user says "GO LOUNGE" and the robot drives there;
- the path planner algorithms avoids the "STAIRS" and positions the robot near "ELEVATOR BUTTON";
- a health monitoring component queries the manager of tags for an `Object3D` with a "NURSE" tag;
- the battery status monitor along with the wheelchair weigth monitor issue a method for the robot to drive to the "DOCK" station to recharge itself;

In order to be used between different applications the tags provide a discrete classification of the objects. Our approach is just qualitative, although components can modify the tags of objects based on any classification or fuzziness algorithms they implement.

Manager of zones This component is responsible for delimiting polygonal zones in the robot representation of the world. `Zones` derive from `Object3D` and so they also have `Tags`. This component provides the following services :

- get the center of a zone that contains a certain tag or combination of tags;
- ask if a certain point lies inside the desired zone polygon to answer the "are we there yet" question that path planners may query;
- geometry operations like area and perimeter of the zones;
- serialize a `Zone` in XML and describe it to other components;
- edit zones, either from a XML message from other components or with GUI input;

Spatial awareness This component monitors spatial relations between the entities in the environment, i.e. distances between `Object3D` or if `Object3D` are inside `Zones`. The idea is to include inside simple "if" conditions relative to positions between the environment entities and execute methods if they trigger. Other components register XML messages to be delivered to any components if events trigger. In order to trigger events the component periodically checks the positions of the monitored entities.

Situational awareness This component monitors the environment for generic events, that happen in the form of new/deleted `Object3D` or the addiction or removal of `Tags` from `Object3D` entities. An example is a perception component that will create a "PERSON" `Object3D` using the manager of tags, the manager of tags announces this new `Object3D` to any components listening for new `Object3D` events. A follow behavior component can then launch, and the "FOLLOW" prefix will be loaded into the task context. Another more pervasive example is the monitorization of the health of a person, that in case the person changes its health from "HEALTHY" to "SICK", the component can launch an emergency management component to handle the situation.

Interaction based on FSG

FSG are the least common denominator between a speaking-enabled, illiterate, person and a computer/device. Therefore by enabling a FSG communications such as speech we are maximizing the range of potential users of our system. The FSG layer of components enable a user to communicate with any number of any devices, which is just by itself a progress since most robotics research that involves speech just uses the input to position the robots. Next are described the components of this layer: the inputs, the context selector and the task context.

FSG Inputs FSG inputs can come from speech recognition, keyboard input, or from a scheduler (for scripting).

A Player speech recognition proxy (also contributed by us) is used in order to obtain utterances from the speech.

The scheduler has the option of sending commands at specific times in the future. When acting like a FSG recorder, the scheduler requires a context selector to select it, so that the user can direct all FSG commands exclusively to the recorder component.

These text inputs are formatted in XML to ease parsing by sequent input components.

Context selector and ignore mode This component does two things: 1) turn on and off the ignore mode so that the user can speak almost freely (except for the word that starts listening again); 2) establish delivery channels from the XML inputs to task context(s). The selected task contexts will receive the XML formatted text.

The selection of task context can be optional if we wish to work with only one task context (one robot), in this case there is no need to specify the strings to select and reset the contexts. Support for multi-casting the commands to multiple robots is done using the words "ALL" or adding together the ids of components using the conjunction "AND" word.

The "SELECT" and "IGNORE" words are exclusive to the context selector, i.e. words that cannot be used in the tasks context if the context selector component is in use. Although the user can configure which words he wants to use for selection and ignoring commands in the configuration file.

Task context The task context component holds a list of strings that are the prefix of command phrases. A prefix of a command phrase $T = t_1 \ldots t_n$ is a phrase $\widehat{T} = t_1 \ldots t_m$, where $m \leq n$. When the incoming prefixes matches a prefix registered to a component, the whole command is delivered to that component. As example of this, the navigation component has the "GO TO", "TURN", "STOP" prefixes registered, the X10 (home automation protocol) component has "LIGHTS" or "DOOR" prefixes registered.

Normally commands to robots are given in the form of verbs, but because we do not want to limit the way the system works FSG can have any form of syntax.

Speech Contexts are dynamic, in the sense that other components can at any time register their prefixes. When components register their prefixes they receive an unique id in order to remove that prefix later from the context. Also external components can check the list of registered prefixes and components to avoid conflicts that could result in future hazards.

The interpretation of the meaning of FSG is only done in the task components. One advantage of postponing the interpretation of utterances to the task components is in avoiding the interpretation of phrases representing quantities to components that do not make use of them.

The concept of context has a broader application than each robot having only one context. The robot can have more than one context, where each context corresponds to a task the robot can perform.

Robot tasks and behaviors

There are at least two ways of writing robot tasks:

1. Write a dedicated component for the task, this will enable a fine grained personalization of how the task is achieved. It allows the C++ language to control the sequence of events. An example of this kind of task is the "WAITRESS" component.

2. Combine existing components functionality with the events of the cognitive layer to describe new tasks. In this case the flow of control will be limited by the possibilities offered by other components. This has the advantage of not having to understand C++ language, because the behavior is defined by editing the XML messages between components. An example of this is the augmented navigation task.

During initialization the components load prefixes into the task context component, so they can be addressed using the FSG. As an example the "WAITRESS" component adds the "DELIVER" prefix and the the navigation component of the augmented navigation task adds typical navigation verbs like "TURN", "GO", "FORWARD", etc.

Grab and delivery This example component waits for orders to grab an `Object3D` in a zone and a delivery to another `Object3D`. It subscribes a Player position proxy in order to carefully approach the item to grab. During the grab and delivery operation it also contacts other ERF components, that are : the manager of zones to query the center of the grab zone; the navigation component to plan the path and drive to the zone; the fiducial to identify the item to grab; the gripper to grab the item; and finally the manager of tags to locate the `Object3D` to deliver the item to.

Augmented navigation The augmented navigation is a combination of a pre-existing navigation component and a spatial awareness configuration. This spatial awareness configuration states that if the `Object3D` "WHEELCHAIR" is inside the "MY ROOM" `Zone`, then it will make available two more zone, the "TABLE" and the "BED".

The navigation component provides access to both a path planner and to a velocity controller that in Player have two distinct proxies and interfaces. It needs access to both because the commands delivered to both Player proxies produce actions that are not mutually exclusive, like the "STOP" that must be given to both proxies simultaneously.

The internal mechanism of the navigation component is a state machine, where each command will deliver the mechanism to a corresponding state. Normal position interface commands like "TURN" and "FORWARD" work as expected. An operation that involves other ERF components is the "GO" command that for parameter has the tag of a `Zone` in the environment. When the "GO" is issued this component contacts the manager of zones to query for the center of a zone with that tag.

Robot world and Player proxies

Our voice recognition system uses the library CMU Sphinx [CMU, 2001] which means the system is speaker independent, and does not require training. The robot navigation algorithm VFH [Konolige, 2000] and the path planner Wavefront are both provided by Player [Gerkey et al., 2003]. Player also let us work either in simulation or in the real-world.

Example of usage

In this section is demonstrated one use case of a user interacting with multiple robots. The scenario serves to demonstrate how the concepts exposed in this paper work together in a unified framework for interaction and cognition. The scenario is of a nursing home with two actor robots. One is a service robot called "WAITRESS" and the other is the patient "WHEELCHAIR".

Because our system does not require the real-world and the simulation simplifies the reproduction of the experiment we use the Stage simulator [Gerkey et al., 2003] to provide us with a virtual world with robots and grabable items that the robots can interact with.

The script of action consists in an a patient in the nursing home, before meal time, when he ask the "WAITRESS" robot to grab him a meal, while he goes to his "ROOM", to the "TABLE", before meeting the service robot in the "LOUNGE", finally he takes the service robot to his "ROOM", to help him with other chores.

Proc. LangRo'2007
978-972-96895-2-9

The order of voice commands and the final visual log of the experiment are depicted in Fig.2. The experiment is executed as follows:

1) First the user selects the context (in this case a robot) with id "WAITRESS" and commands it to get a "PIZZA" from the "KITCHEN"and deliver it to him. 2) The user selects his "WHEELCHAIR" and asks it to go to "MY ROOM". 3) Then is tested the capacity to ignore commands so the user can speak almost freely with other persons. The robot starts ignoring commands when the user says "IGNORE" and restarts listening when he understands "AWAKE". This way the user can be more confortable speaking with other persons until he is ready to interact again. 4) When the "WHEELCHAIR" arrives at the "ROOM" zone, new zones appear automatically, namely the "TABLE" and "BED". This was possible because the spatial awareness component was monitoring the presence of the "WHEELCHAIR" `Object3D` inside the zone "ROOM" and when that event triggered it sent a XML message to the manager of zones to create the new sub-zones, and another message to the speech recognition proxy to add the words "TABLE" and "BED" to its recognition dictionary. This way the speech recognition confusion matrix is reduced by only working with with the words possible in the current task context. Another advantage is to just load new components and data when they are needed, depending on the spatial context. 5) the user then goes to the "LOUNGE" where the "PIZZA" is delivered to him by the "WAITRESS". 6) Finally we select both robots simultaneously and asks them to go to the "ROOM". The experiment ends.

Future work

This paper contributes with a new branch of functionalities to the ERF trunk, which still can be improved in many ways:

- for a better cognitive interaction with space, the robot should be able to better perceive itself by means of introspecting the sensors and actuators it has, and the data the algorithms produce, in order learn to perform new actions with its hardware;
- allowing the robot to create simulations based on the current perception it has of the world, so it can "imagine" better solutions for problems;
- it should be interesting to study the possibilities of cognitive architectures such as SOAR [Laird et al., 1987] in order to execute the two previous suggestions. This can be accomplished by creating a component that acts as a bridge between our light-weight cognitive architecture to the SOAR one. The SOAR architecture can solve problems in the world that involve state changes, like opening doors, sorting sequences of tasks to optimize functions, and other "sokoban"[1] like problems.

[1] a transport puzzle game in which the player pushes boxes around a maze, and tries to put them in designated locations.

Extend the scheduler with a long-term plan of enabling oral programming the components via FSG inputs. This research can be ignited with the following items:

- accept verification of task results, with tickets to check the task status;
- explore with minimal atomic primitive tasks that can be repeated and combined to form high level tasks;
- allow the recorder to understand variables and define methods, that are the result of grouping primitive tasks. An advantage of this will be that by defining the variables along with the FSG, task components like the grab and delivery component, can be made redundant as they can be represented by a set of primitive tasks.

Suggestions that are easier to implement are:

- explore programming conditions using FSG, starting with "IF" and "WHILE";
- implement text completion on the keyboard input;
- study the advantages of having properties, i.e. a data tuple that maps a text entry to any entry. Properties which are mostly quantitative description can be harder to parse but are more precise descriptions of the world than tags;
- separate the cognitive layer into a global for all the robots, and a local one exclusive to the robots the user interacts with, to handle properties like "MY ROOM" that in a global layer would be "$USER ROOM".
- transpose the components to the linux desktop, so that applications can take advantage of speech and text interaction, e.g. selecting pixels in image manipulation programs and say "CUT", "COLOR BLUE", or in text applications to format text like saying for example "FORMAT BOLD". Also it would be interesting to have a robot interacting/talking with a desktop computer or other home appliances speaking between them, either via voice or by exchanging XML messages. Note that this is not an attempt at solving the Turing test;
- plan a variety of use cases that will help to emerge some canonical concepts relative to world description and interaction.

Contributions and conclusion

In this work is introduced an interaction platform for one user to multiple robots. This platform is multi-modal, and in this article we describe our implementation of the FSG inputs and how it copes with a light-weight cognitive layer for describing and interacting with the robot environment. The approach of using a dynamic lexicon based on FSG, and the resulting advantages over traditional static lexicons are discussed.

We describe the design and implementation of the building blocks that are implemented as components, and that can be put together to achieve versatile FSG based

interactions. We explain the design of the awareness contexts that are essential for the spatial and the situational awareness of the platform.

In an use case is demonstrated how all these features are combined seamlessly together to create a dynamic composed human robot interface.

Users and researchers can assign tasks and behaviors to robots from speech, keyboard, or record scripts of actions using FSG. Components that execute these tasks and behaviors wrap low-level algorithms and robot drivers in order to execute the tasks. These tasks can then be applied to many robots at once. This feature allows researchers from other areas that do not deal with low level programming languages to use our software also, as they only have to learn the high-level operation of components.

The software is available on the Internet for free, in the miarn [Xavier, 2007] site.

References

Birk, A. and Pfingsthorn, M. (2006). A human machine interface supporting adjustable autonomy of rescue robots. In Noda, I., Jacoff, A., Bredenfeld, A., and Takahashi, Y., editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 267 – 278. Springer.

Chong, S., Kuno, Y., Shimada, N., and Shirai, Y. (2000). *Advances in Multimodal Interfaces - ICMI 2000: Third International Conference*, chapter Human-Robot Interface Based on Speech Understanding Assisted by Vision, page 16. Springer Berlin / Heidelberg.

Clark, C. and Frew, E. (2003). An integrated system for command and control of cooperative robotic systems. In *International Conference on Advanced Robotics*.

CMU (2001). Cmu sphinx speech recognition engine. Available online at cmusphinx.sourceforge.net.

Flippo, F. (2003). A natural human-computer interface for controlling wheeled robotic vehicles. Available online at citeseer.ist.psu.edu/flippo03natural.html.

Fong, T., Kaber, D., and et al. (2004). Common metrics for human-robot interaction. Available online at citeseer.ist.psu.edu/fong04common.html.

Gerkey, B., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proc. 11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal.

Graves, B. (1995). A generalized teleautonomous architecture using situation-based action selection.

Halme, A., Jakubik, P., Schonberg, T., and Vainio, M. (1995). Controlling the operation of a robot society through distributed environment sensing. In *IFAC-Workshop on Human-Oriented Design of Advanced Robotics Systems*, Vienna, Austria.

Jones, H. and Hinds, P. (2002). Extreme work groups: Using swat teams as a model for coordinating distributed robots. In *Conference on Computer Supported Cooperative Work*.

Konolige, K. (2000). A gradient method for realtime robot control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Kuipers, B. (2000). The spatial semantic hierarchy. *Artif. Intell.*, 119(1-2):191–233.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: an architecture for general intelligence. *Artif. Intell.*, 33(1):1–64.

LaValle, S. and Ku, J. (1999). Randomized kinodynamic planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Detroit, Michigan, USA.

Levine, S., Bell, D., Jaros, L., Simpson, R., and Koren, K. (1999). The navchair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*.

Makarenko, A., Kaupp, T., Grocholsky, B., and Durrant-Whyte, H. (2003). Human-robot interactions in active sensor networks.

Nourbakhsh, I., Sycara, K., Koes, M., Young, M., Lewis, M., and Burion, S. (2005). Human-robot teaming for search and rescue. Available online at citeseer.ist.psu.edu/nourbakhsh05humanrobot.html.

Pires, G. and Nunes, U. (2002). A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. volume 34, pages 301–314.

Scholtz, J., Antonishek, B., and Young, J. (2004). Evaluation of a human-robot interface: Development of a situational awareness methodology. *Hawaii International Conference on System Sciences (HICSS)*, 05.

Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., Adams, W., Bugajska, M., and Brock, D. (2004). Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man and Cybernetics*.

T. Iwase, A. N. and Kuno, Y. (2004). Robotic wheelchair understanding the userś intention in speech using the environmental information. In *Advances in Computer Science and Technology*.

Tellex, S. and Roy, D. (2006). Spatial routines for a simulated speech-controlled vehicle. In *HRI '06: Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 156–163, New York, NY, USA. ACM Press.

Xavier, J. (2007). Modules for intelligent autonomous robot navigation. Available online at miarn.sf.net.

Xavier, J., Pacheco, M., Castro, D., Ruano, A., and Nunes, U. (2005). Fast line arc/circle and leg detection from laser scan data in a player driver. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona.

Zelek, J. S. (1997). Human-robot interaction with a minimal spanning natural language template for autonomous and tele-operated control. In *International Conference on Intelligent Robots and Systems*.

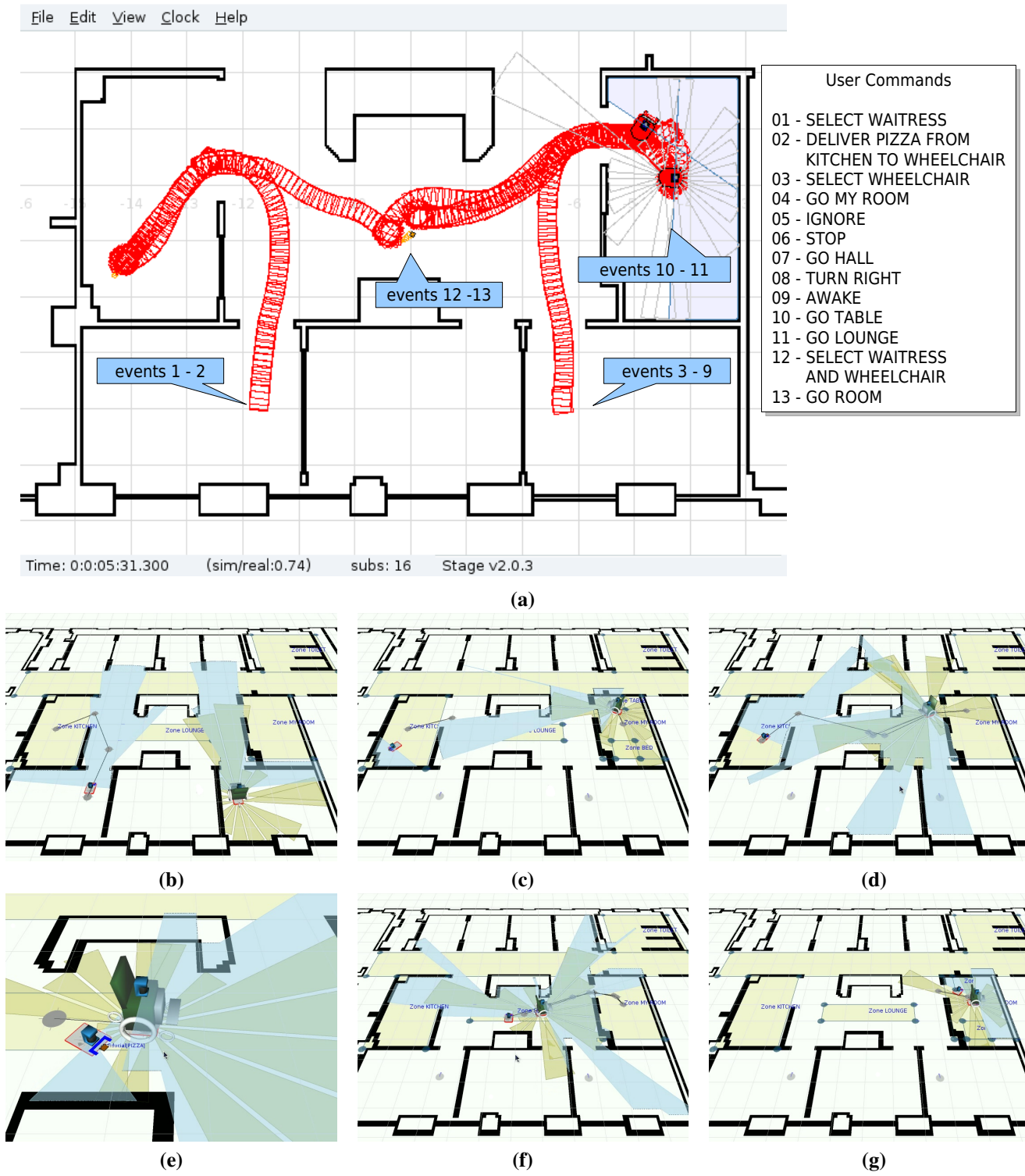


Figure 2: The experiment conducted in the stage simulator controlling two robots, the tracks of the robots and the commands that originated them are listed by order in a). In b) the WAITRESS robot is asked to grab a pizza from the kitchen and deliver it near robot WHEELCHAIR; in c) the WHEELCHAIR has entered the ROOM and because the spatial awareness layer was monitoring the "MY ROOM" with instructions for augmentating it, new zones, TABLE and BED are now available; in d) the WHEELCHAIR leaves the room, the augmented zones disappear; in e) WAITRESS robot is near the WHEELCHAIR so it delivers the PIZZA; in f) the user asks both robots simultaneously to "GO MY ROOM"; in g) the experiment ends in the room of the user, once again with the augmented zones.

Proc. LangRo'2007
978-972-96895-2-9