Symbol Grounding through Cumulative Learning

Samarth Swarup[1], Kiran Lakkaraju[1], Sylvian R. Ray[1], and Les Gasser[1,2]

[1] Dept. of Computer Science,
[2] Graduate School of Library and Information Science,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801, USA
{swarup, klakkara, s-ray1, gasser}@uiuc.edu

Abstract. We suggest that the primary motivation for an agent to construct a symbol-meaning mapping is to solve a task. The meaning space of an agent should be derived from the tasks that it faces during the course of its lifetime. We outline a process in which agents learn to solve multiple tasks and extract a store of "cumulative knowledge" that helps them to solve each new task more quickly and accurately. This cumulative knowledge then forms the ontology or meaning space of the agent. We suggest that by grounding symbols to this extracted cumulative knowledge agents can gain a further performance benefit because they can guide each others' learning process. In this version of the symbol grounding problem meanings cannot be directly communicated because they are internal to the agents, and they will be different for each agent. Also, the meanings may not correspond directly to objects in the environment. The communication process can also allow a symbol meaning mapping that is dynamic. We posit that these properties make this version of the symbol grounding problem realistic and natural. Finally, we discuss how symbols could be grounded to cumulative knowledge via a situation where a teacher selects tasks for a student to perform.

1 Introduction

Where do meanings come from? This is one of the most important questions underlying the study of cognition, language, and artificial intelligence. In the field of artificial intelligence, the intellectual history of this problem traces back to the earliest speculations on the nature of intelligence [3]. Alan Turing, in the conclusion to his classic article which introduced the Turing test, suggested that there might be at least two routes to building intelligent machines: attempting very abstract activities like playing chess, or outfitting a computer with sensory devices and then attempting to teach it natural language [1]. In subsequent years the purely symbolic approach gained dominance, partly due to the comparative ease of building purely symbolic systems, and partly due to the influence of the

[3] It should be pointed out that this question has a much longer history in philosophy. See, e.g., [2] for a review.

Physical Symbol System Hypothesis of Newell and Simon [3], which says that a set of symbols, combined with appropriate rules for their manipulation (essentially, a *formal system*), is sufficient for general intelligent action. The implicit assumption underlying this view is that intelligent behavior from a machine does not require that the machine "understand" things in the same way as we do.

In response, Searle argued, using his famous Chinese Room Argument, that there is a distinction between intelligent *behavior* and true intelligence [4]. A person could undertake the Turing test in a language unknown to him (say, Mandarin), if he possessed an appropriate program. This program would be a set of rules for manipulating symbols in Mandarin, which he would use to transform questions into answers, and thereby pass the Turing test (if the rules are good enough). Since the person does not know Mandarin, the symbols have no *meaning* for him, though it would appear so to the observer. Intuitively, it seems, a machine using this program would not be *truly* intelligent.

In an attempt to bridge this gap between a symbolic system and a truly intelligent system, Harnad formulated the *symbol grounding* problem. In his words, the problem is thus, "How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads?" [5]. Though this problem arose in the context of the limitations of purely symbolic systems in cognitive modeling, it was realized to be of fundamental importance in the study of language evolution and the design of artificial languages. Symbol grounding, in this context, concerns the problem of relating the conceptualizations underlying a linguistic utterance to the external world through an agent's sensori-motor apparatus [6].

Harnad suggested that the symbol grounding problem could be solved by building a hybrid symbolic-nonsymbolic system in which symbolic representations are grounded bottom-up in non-symbolic representations which are either *iconic* or *categorical.* Iconic representations correspond directly to objects and events, and categorical representations are based on generalizations from iconic representations (i.e. concepts such as "animal", which do not have direct real-world analogs). This highlights one very important aspect of the symbol-grounding problem: it is concerned with ontology construction. However it ignores another, equally important, aspect: a symbol is a convention between two (or more) agents. Thus it makes no sense for a single agent to try to ground symbols. Further, ontology construction and the construction of a corresponding symbolic system (i.e. lexicon acquisition) are inter-dependent. A new symbol might be created for a new ontological category. Conversely, a new ontological category may be created in response to the use of a symbol by another agent.

This interdependence between symbols and meanings has been understood and incorporated in subsequent work on lexicon acquisition and symbol grounding, most clearly in the well-known series of Talking Heads experiments. See [7] for a review of these and other experiments based on language games. The main issue we have with these experiments is that they consider the development of a shared lexicon to be the *primary* task in which the agents are engaged. Thus, in these experiments, meanings are created primarily through the process of the language game. The argument of this paper, however, is that meanings should be derived from the tasks that a cognitive agent is faced with in the course of its lifetime. Otherwise they will have no relevance to the agent. In other words, the agent will have the means but not the need to communicate. In what follows, we outline a method for combining the processes of solving multiple problems, and developing a grounded symbolic communication system to aid problem-solving.

We first discuss the process of ontology construction, and how an ontology might be extracted from the process of learning to solve multiple related problems. We call this process *cumulative learning*, because the knowledge extracted from the tasks accumulates over time. Since each agent extracts its own cumulative knowledge, these meanings are entirely internal to the agent. We then discuss how it might be possible to ground symbols to this cumulative knowledge, followed by a discussion of some of the consequences of this process. In the concluding section, we discuss some of the advantages and limitations of our approach, and possible future work.

2 Ontology Construction

An ontology determines the domain of discourse, i.e. what a language talks about. From the point of view of an agent, these are the entities that are relevant to the problems or tasks with which it is confronted. The ontology of an agent in a mushroom world, e.g., might contain types of mushrooms, features (such as color, size, and shape) by which these might be distinguished, etc. It might also contain more abstract concepts, like "edible", "poisonous", etc. [8]. Some of these ontological entities might be pre-specified, the result of processes like biological evolution or engineering design. Other entities would be discovered by the agent as it learns to perform the task of distinguishing edible from poisonous mushrooms. This is a primary task for the agent, and each agent could try to solve this task in isolation. However, they clearly stand to gain by developing a language to communicate about these concepts:

- an agent, Alice, who is proficient in distinguishing edible mushrooms from inedible ones, might communicate to another agent, Bob, whether a particular mushroom is edible,
- Alice might be able to teach Bob to distinguish edible mushrooms from poisonous ones himself, assuming he has the same ontology,
- Alice might be able to help Bob acquire the necessary ontological categories for distinguishing edible mushrooms from poisonous ones.

The point is that the ontology emerges from the primary task, though its acquisition might be facilitated by the secondary task of language acquisition. Thus the meanings in a lexicon must have some functional significance for the agent. This is an aspect of language evolution that is missing from most previous work on symbol grounding, with some exceptions [8, 9].

Over its lifetime, an agent is expected to encounter many tasks, which might be related to each other. Ideally, the agent should not just learn to solve each task, but should also *learn how to learn*. In other words, if the tasks are related, the agent should be able to improve its learning performance, exhibiting quicker and more robust learning on each new task. This is a subject of much research in the machine learning community and is known variously as transfer learning, multi-task learning, lifelong learning, etc. Generally the improvement in learning performance is achieved by using some learnt information, such as invariants, priors, etc., to bias the learning of new tasks. Our suggestion here is that this learnt information, which we call *cumulative knowledge*, could form the ontology of the agent. We discuss this in more detail below.

3 Cumulative Learning

We use the term cumulative learning to refer to the case where an agent explicitly accumulates a store of knowledge that is extracted from solving multiple tasks and is useful for solving new tasks. The key issue in cumulative learning is that of recognizing, and exploiting, similarities between tasks. That human language is efficacious in this process is suggested by studies of analogical thinking in problem solving [10]. In fact, it has been argued that analogy-making is the core of cognition [11].

A cumulative learning system consists of two parts: a learning mechanism and a knowledge extraction mechanism. These two mechanisms could conceivably use two different representations: effectively a task-dependent, and a task-independent representation, e.g. the learning mechanism could be a recurrent neural network, and knowledge could be extracted in the form of finite state automata [12]. People have also attempted to combine feed-forward neural networks with symbolic rules [13]. However, the drawback to these approaches is that there is always the possibility of translation noise. A recurrent neural network, e.g., is capable of embedding some context-free and context-sensitive grammars [14], and therefore attempting to represent the learnt recurrent net as a finite-state automaton might create errors.

Other approaches attempt to directly transfer parts of the learned neural network, such as the first layer of weights. The idea is that these might represent features that are useful for multiple tasks [15, 16]. The limitation of this approach is that knowledge transfer is only possible within-domain, because if the neural networks do not have the same dimension, it is not possible to reuse the weights.

To get around these two problems, we have presented a cumulative learning method that uses graph-structured representations [17]. Learning is done with a genetic algorithm, and knowledge is extracted by mining frequent subgraphs. The idea is that these frequent subgraphs can be used as primitives by the genetic algorithm in the construction of candidate solutions for new tasks, thereby learning faster. Since these networks do not have fixed dimension, we avoid the inflexibility of neural networks. We tested this idea on a set of Boolean function domains. The domains are parameterized by their dimension, n, and the tasks are parameterized by the number of adjacent 1's, k, that must be present in the input for a positive example. For example, a task might consist of inputs of dimension four, where an input is classified as positive if two adjacent 1s appear in the input vector. We name this task 4inputs-2adj-ones.

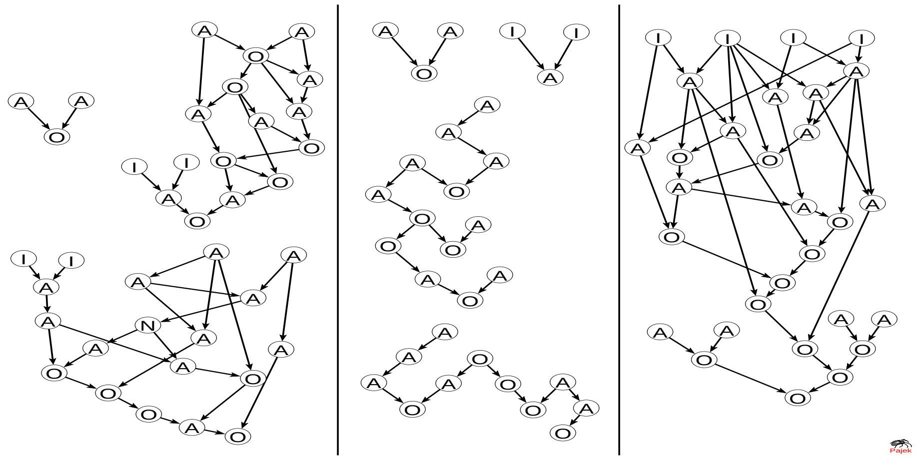

Fig. 1. The left box shows networks learned on the first three tasks (4inputs-2adj-ones, 8inputs-2adj-ones, and 8inputs-3adj-ones). The middle box shows 4 of the 18 subnetworks, extracted by the graph-mining algorithm CloseGraph [18], which appear in at least two of the networks on the left. These would constitute the "meanings" for symbolic communication. The right box shows the network learned for the 12inputs-2adj-ones task, in which some of the subnetworks are seen to appear several times.

Initially, the agents have a very small set of primitives, consisting of just single nodes that compute the AND, OR, and NOT functions. During the cumulative learning process, they extract many more primitives which are small networks that can be combined together to solve many tasks. This is illustrated in figure 1. The left box in the figure shows the networks learnt on the first three tasks: 4inputs-2adj-ones, 8inputs-2adj-ones, and 8inputs-3adj-ones. AND nodes are labeled A, OR nodes are labeled O, NOT nodes are labeled N, and nodes labeled I are "input" nodes which copy their input unchanged to their output. A and O nodes are assumed to take two inputs, and N and I nodes are assumed to take one input. All arrows in the figure point downward. If a node has fewer inputs shown than it is assumed to require, the remaining inputs are to be supplied externally (i.e. from the input vector).

The middle box in the figure shows some of the sub-networks extracted by the CloseGraph algorithm [18]. These sub-networks are used by the genetic algorithm as primitives when learning to solve the next task: 12inputs-2adj-ones. The right box shows the network learned for this task. Some of the sub-networks are seen to appear in this new network, either whole or in part (where they have been incorporated and then further mutated).

Though this is a very artificial set of tasks, the same kind of representation and cumulative learning method could be used in a more realistic setting, such

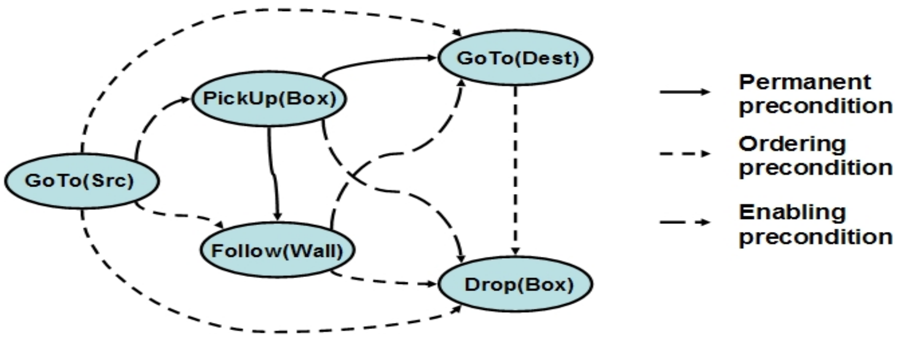

Fig. 2. An example behavior network. Reproduced from [19].

as motion planning with robots. The idea of primitives carries over easily to this domain, and "behavior networks" (see fig. 2) have been used to represent motion plans [19].

Figure 3 shows a typical comparison of learning curves with and without transfer of knowledge from previously learnt similar tasks. Knowledge transfer results in both faster learning and reduced variance in error (i.e. more robust learning). If another agent, who already knows how to solve the given problem, is able to tell the agent which primitives to use to solve the task, learning would converge even faster.

A natural follow-up to the idea of cumulative learning, therefore, is that the extracted cumulative knowledge might constitute the ontology or meaning space of the agent. To use a somewhat provocative term, an agent *understands* a new task in terms of its cumulative knowledge. The challenge, then, is to develop a symbol system which maps onto the agents' cumulative knowledge and enables communication that helps in learning. We posit that this is a very natural and realistic version of the symbol grounding problem for the following reasons:

- In this setup, meanings are internal to the agents and are not (indeed cannot be) directly communicated. Direct meaning communication is a problem with a lot of the previous work on symbol grounding, as various researchers have begun to point out [20], and work around [21, 22].
- Different agents may (and probably will) have somewhat different sets of internal meanings, depending on the sets of problems they have encountered. This is both a problem to be surmounted, and a realistic feature of our setup. We address this further in the next section. Previous work on language evolution has often assumed that all the agents have the same fixed set of meanings. There have been a few notable exceptions, such as [20, 23].
- Meanings may not necessarily correspond to objects in the environment. The common example is, what is the meaning of the symbol "chair"? It seems that "chair" corresponds to some prototypical chair which only exists in our minds, and not necessarily in the environment. In the same way, the

extracted cumulative knowledge may not necessarily correspond to objects in the environment, though if the same objects are encountered sufficiently often, they might be represented in the cumulative knowledge.

– Another advantage of our setup is that the agents could acquire symbols not just for objects (i.e. nouns and adjectives), but also for actions (i.e. verbs), or perhaps more abstract concepts.
– Last, but not least, in this setup the symbol-meaning mapping does not have to be static. This is because the context imposes equivalences on items of cumulative knowledge and allows a symbol that normally refers to one item to be interpreted as referring to another item. We address this point in more detail in the next section also.

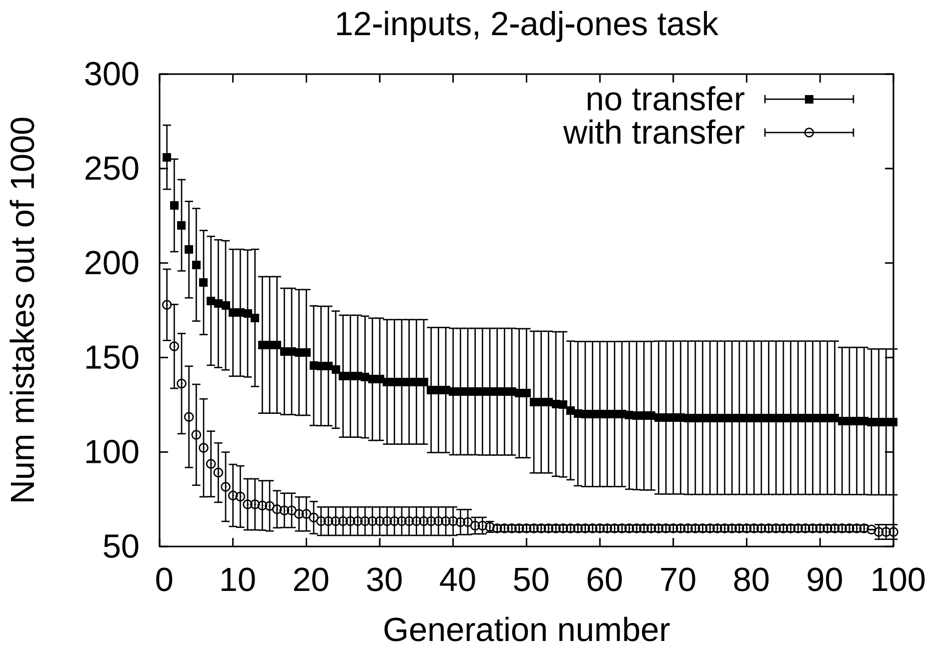

Fig. 3. A typical comparison of learning performance with and without transfer of knowledge. Transfer of knowledge from similar tasks results in faster and more robust learning.

4 Learning to Ground Symbols to Cumulative Knowledge

We now address the question: what might be a learning procedure that leads to the development of a communication system based on cumulative knowledge?

The underlying goal is to create a language that is useful in learning new tasks. Thus the development of the communication system should be guided by learning performance. Suppose Alice knows how to solve a task, i.e. she knows which of *her* internal primitives she can compose to create a solution to a particular task. If she could communicate this information to Bob (and assuming he has the same internal primitives), Bob could learn to solve that particular task essentially in one step.

There are at least two major hurdles to grounding symbols to cumulative knowledge. The first is that cumulative knowledge is entirely internal to the agent. There is nothing the agents can point to, nor can we use the cross-situational learning strategy of [20, 22]. The second problem is deeper: since meanings depend on the tasks that have been encountered by the agent and the solutions that the agent has discovered, there is a danger that the meanings may be too different to allow successful communication. In other words, we need some mechanism to keep the ontologies aligned.

To surmount these two problems, we suggest a parent-child (or teacher-student) scenario. This is not an unrealistic assumption, and has been made before, e.g. in the Iterated Learning Model [24]. The parent selects tasks for the child to perform. This gives the parent some measure of control over the ontology that the child is likely to develop. Let us suppose, for simplicity, that the agents are learning Boolean functions. The parent selects a task for the child by creating a training set of labeled examples, which the child must now learn to classify. The child initially has a small set of primitives. Let us assume that the child knows how to compute the AND, OR, and NOT functions. The parent and child, however, do not have a shared symbol system corresponding to these primitives. The parent can easily establish a set of symbols for these functions by selecting extremely simple tasks for the child to perform. For example, if the first task is to perform the AND operation on two inputs, the parent presents the child with the training set: $\{\{(0, 0), 0\}, \{(0, 1), 0\}, \{(1, 0), 0\}, \{(1, 1), 1\}\}$, and the symbol `AND`. The symbol is meant to indicate which primitive the child should use to solve the task. The child does not know to which of its primitives the symbol `AND` corresponds, but attempting to solve the task quickly tells it that there is only one primitive which works. It is, of course, possible that the child happens to generate a more complicated network that computes the AND function, so we might need to assume an in-built bias towards smaller networks in the learning algorithm (similar to Ockham's razor).

In a similar way, the parent and child could develop symbols for the OR and NOT functions also. This small shared lexicon provides a foothold for the development of a more complex ontology and a corresponding symbol system. As the parent selects more complex tasks for the child to perform, they will need to develop a convention for communicating about combinations of primitives. This could be something simple, such as generating a sentence by ordering primitives by the number of times they are used in the task, e.g. "`AND OR`" would indicate that the AND function is to be used more than the OR function in the new task.

As the child learns more complex tasks, its ontology will grow (by mining frequent subgraphs). The parent will not know exactly what the subgraphs that the child has discovered are, but by judicious selection of tasks, it should be possible to guide the emergence of the child's ontology, and to maintain a shared lexicon.

4.1 The Importance of Starting Small

Note that it is very important to *start small.* The parent and child could not develop a shared lexicon if the initial tasks don't serve to bootstrap the communication system. This is rather reminiscent of Elman's work on learning and development with neural networks [25]. He has talked about a rather interesting phenomenon about the learning of natural language: that it is much easier to train neural networks to process natural language sentences if we use a developmental paradigm where initially the networks are severely restricted with respect to their working memory. This essentially focuses the attention of the network on precisely those linguistic structures which help it to subsequently learn the more complex structures (see also [26]). Our symbol grounding procedure suggests that ontology alignment might be a reason why natural languages exhibit this surprising property.

There is a deeper reason for starting small as well. The process of cumulative learning itself benefits greatly from starting small. In other words, even if an isolated agent were learning to solve multiple related problems, it would benefit from starting with small problems. The reason is that for any given problem there are multiple networks that will perform well on it. Not all of them are good from the point of view of knowledge transfer however. One of the problems in cumulative learning, then, is how to find the networks which have subgraphs that can be reused for solving other problems? One solution is to start with small problems, which have very few easy to find solutions. Once the agent starts building up its cumulative knowledge, there is reinforcing effect. Using the cumulative knowledge to find solutions to new problems ensures that more cumulative knowledge will be found. This is a kind of *cumulative advantage* [27].

4.2 Dynamic Symbol Grounding

In real life, symbol grounding has a dynamic or contextual aspect to it. Heidegger refers to this as the "as-ness" of language [28]. In other words, language enables us to see the world (or the context) in a new way. Suppose Alice says to Bob, "I need a hammer." Bob, seeing no hammers around, hands her a rock. This is clearly a successful case of communication, even though the word "hammer" was grounded to a rock by Bob. In fact, Alice's request enabled Bob to see his surroundings in a different way (to see rocks as hammers). This is the "as-ness" that Heidegger is talking about. Our setup also permits dynamic symbol grounding. The task imposes equivalences on the items of cumulative knowledge. This is easy to see with the Boolean function domain. An training set which does not include all possible examples means that there are several Boolean functions which would classify the examples correctly. Further, more complex scenarios might contain situations where not every item in the cumulative knowledge of the agent can be applied, e.g. in fig. 2, some of the preconditions might not be satisfied. The parent might not know which of the child's primitives are inapplicable in a given context, since the primitives of the parent and child will not be identical. Therefore it is easy to imagine situations where the parent suggests using a primitive which the child cannot apply. In such a situation, the child will be forced to interpret the symbol differently, and will apply a perhaps contextually equivalent primitive. The interpretation process can be put in probabilistic terms: the task imposes a prior distribution on which items of cumulative knowledge can be applied, and the child computes a posterior distribution by combining this prior with the suggestion supplied by the parent (which corresponds to a conditional).

5 Conclusion

We have presented our speculations on how a symbolic communication system could be grounded in cumulative knowledge. The advantages of this particular method of meaning construction are several:

- The agents are engaged in multiple tasks over a lifetime, and communication helps in improving performance on these tasks.
- All the agents are not assumed to have the same fixed set of meanings.
- Meanings are internal to the agents, and there is no need for direct meaning communication.
- Symbols and meanings arise in an interdependent manner.
- Symbols can be interpreted in context, i.e. symbol grounding is done dynamically.

There are, however, some limitations to this account as well. One of the main problems is that of ontological structure. Cumulative knowledge, as described here, consists of a set of frequent subgraphs that are useful for multiple tasks. It has no further structure. Ontologies are generally assumed to be hierarchically organized. In this sense, the use of cumulative knowledge as the meaning space of the agent is somewhat unrealistic. On the other hand, it is not clear what could be gained by attempting to give the cumulative knowledge some structural organization. It could be done, however. E.g. if a particular item of cumulative knowledge is a subgraph of another, that implies a `PartOf` relation between them.

Another potential challenge is that of preventing divergence in the meaning space of the agents. The constructive approach we have suggested seems promising in that regard, but it would be sensitive to the "curriculum" chosen by the parent. This bears further investigation. It might also help in answering the question, when does language not evolve?

It might also be argued that not all meanings are cumulative knowledge. We often have names for very specific things, such as "Eiffel tower". This suggests that our account of lexicon development should be combined with other accounts in order to develop a more complete communication system.

Despite these limitations, we believe that there are important connections between cumulative learning and language evolution, and our purpose here is to identify some of these and bring them to the attention of our audience. We believe this can be an area of much fruitful research.

References

1. Turing, A.M.: Computing machinery and intelligence. Mind **49** (1950) 433–460
2. Sowa, J.F.: Signs, processes, and language games. Available Online: http://www.jfsowa.com/pubs/signproc.htm (1999)
3. Newell, A., Simon, H.: Computer science as empirical inquiry: Symbols and search. Communications of the ACM **19** (1976) 113–126
4. Searle, J.: Minds, brains, and programs. Behavioral and Brain Sciences **3** (1980) 417–457
5. Harnad, S.: The symbol grounding problem. Physica D **42** (1990) 335–346
6. Steels, L.: The evolution of communication systems by adaptive agents. In Alonso, E., D.K., Kazakov, D., eds.: Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning. LNAI 2636, Springer Verlag, Berlin (2003) 125–140
7. Vogt, P.: Language evolution and robotics: Issues in symbol grounding and language acquisition. In: Artificial Cognition Systems. Idea Group (2006)
8. Cangelosi, A., Harnad, S.: The adaptive value of symbolic theft over sensorimotor toil: Grounding language in perceptual categories. Evolution of Communication **4**(1) (2000) 117–142
9. Vogt, P.: Anchoring symbols to sensorimotor control. In: Proceedings of the 14th Belgian/Netherlands Artificial Intelligence Conference, BNAIC'02. (2002)
10. Dunbar, K., Blanchette, I.: The inVivo/inVitro approach to cognition: The case of analogy. Trends in Cognitive Sciences **5** (2001) 334–339
11. Hofstadter, D.R.: Analogy as the Core of Cognition. In: The Analogical Mind: Perspectives from Cognitive Science. The MIT Press (2001) 499–538
12. Omlin, C.W., Giles, C.L.: Extraction of rules from discrete-time recurrent neural networks. Neural Networks **9**(1) (1996) 41–52
13. Shavlik, J.: Combining symbolic and neural learning. Machine Learning **14**(3) (1994) 321–331
14. Bodén, M., Wiles, J.: Context-free and context-sensitive dynamics in recurrent neural networks. Connection Science **12**(3,4) (2001) 197–210
15. Thrun, S., Mitchell, T.: Lifelong robot learning. Robotics and Autonomous Systems **15** (1995) 25–46
16. Caruana, R.: Multitask learning. Machine Learning **28** (1997) 41–75
17. Swarup, S., Ray, S.R.: Cross domain knowledge transfer using structured representations. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI), Boston, MA, USA (2006)
18. Yan, X., Han, J.: CloseGraph: Mining closed frequent graph patterns. In: Proceedings of the 9th ACM SIGKDD conference on Knowledge-Discovery and Data Mining (KDD 2003). (2003)
19. Nicolescu, M.N., Matarić, M.J.: Task Learning Through Imitation and Human-Robot Interaction. In: Models and Mechanisms of Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions. (2005)

20. Smith, A.D.M.: Mutual Exclusivity: Communicative Success Despite Conceptual Divergence. In: Language Origins: Perspectives on Evolution. Oxford University Press (2005) 372–388
21. Smith, A.D.M.: Stable communication through dynamic language. In: Proceedings of the 2nd International Symposium on the Emergence and Evolution of Linguistic Communication. (2005) 135–142
22. Beule, J.D., Vylder, B.D., Belpaeme, T.: A cross-situational learning algorithm for damping homonymy in the guessing game. In Rocha, L.M., Bedau, M., Floreano, D., Goldstone, R., Vespignani, A., Yaeger, L., eds.: Proceedings of the Xth Conference on Artificial Life, Cambridge, MA, USA, The MIT Press (2006)
23. Steels, L., Vogt, P.: Grounding adaptive language games in robotic agents. In Husbands, C., Harvey, I., eds.: Proceedings of the Fourth European Conference on Artificial Life, Cambridge and London, MIT Press (1997)
24. Smith, K., Kirby, S., Brighton, H.: Iterated learning: A framework for the emergence of language. Artificial Life **9**(4) (2003) 371–386
25. Elman, J.L.: Learning and development in neural networks: The importance of starting small. Cognition **48** (1993) 71–99
26. Clark, A., Thornton, C.: Trading spaces: Computation, representation and the limits of uninformed learning. Behavioral and Brain Sciences **20**(1) (1997) 57–67
27. de Solla Price, D.: A general theory of bibliometric and other cumulative advantage processes. Journal of the American Society for Information Science **27** (1976) 292–306
28. Heidegger, M.: On the Way to Language. Harper, San Francisco (1982)