Artificial Intelligence Review **10**: 83–102, 1996.

Successful Naïve Representation Grounding

N.J. SALES, R.G. EVANS AND I. ALEKSANDER*
Neural Systems Engineering Research Group,
Department of Electrical and Electronic Engineering,
Imperial College of Science, Technology and Medicine,
London SW7 2BT, U.K.
E-mail: cyber@ic.ac.uk,† *rgevans@ic.ac.uk, i.aleksander@ic.ac.uk*

Abstract. A Neural State Machine, constructed from weightless artificial neurons, is used to "ground" visual and linguistic representations, from real-world sources, using minimal pre-processing. Simulations are used to demonstrate that the system can work in real time, in a real environment, to provide a means of grounding both concrete nouns, and some simple motion verbs.

Key words: Weightless Artificial Neural Networks, representation grounding, multi-modal integration

1. Introduction

This paper addresses the problem of representation grounding from a novel perspective: in it, a naïve, associationistic approach to the association of visual and linguistic representations for integrated processing will be presented. The technology used, Weightless Artificial Neural Networks, allows this unique approach to be applied simply and very quickly on visual data from real grey-scale video, and a variety of linguistic data sources. The paper will show that useful and practically applicable results can be achieved with the minimum of data pre-processing, and the use of a quite general network design.

Some examples of so-called "symbol grounding" tasks will be given, to demonstrate the practical applicability of the work: simulated neural networks will be shown to be able to learn non-trivial associations between visual input and linguistic input, in a relatively unrestricted visual environment, both for concrete nouns, and movement verbs. Potential problems associated with

* The MAGNUS simulation software was produced as a result of a SERC research project: GR/F 34756. The SERC/EPSRC are funding the present research project: GR/J 07228. The authors would like to thank the other members of the MAGNUS project: T. J. W. Clarke, Panayotis Ntourntoufis, John Stonham, and Manissa Wilson. The anonymous reviewers also did a fine and helpful job, for which we are very grateful.

† Corresponding author.

the "naïve" approach will then be discussed. Specifically, the consequences of using no non-trivial pre-processing on input will be explored, and several potential problems will be seen to arise. It will be argued that these "problems" are not problems at all, however, but perfectly acceptable consequences of the system's naïvety and practicality.

If the foregoing discussion has made the approach advocated here sound like a trivial one, that impression will be challenged at the end of the paper, where it will be argued that, despite the lack of computational complexity in the system, the application of Neural State Machine analysis can allow a seamless transition from the non-symbolic processing it carries out, to a symbolic (state machine-like) domain, which could be of great value in the creation of Neural-AI hybrid systems.

In the remainder of this paper, we shall identify why our approach is "naïve" in comparison with other approaches, briefly introduce the neural network technology used in conjunction with it, and discuss dynamic networks and Neural State Machines. We will then present simulation results for the grounding of both concrete nouns and action verbs, and finally fend off the strongest criticisms of our approach which have appeared so far.

2. Grounding representations of language

The most significant question for any approach to automated Natural Language Processing (NLP) is how it might be possible to non-trivially connect the representations of language in an NLP system with the meanings we, as users of natural language, expect them to have. The non-triviality of these connections is paramount: it is perfectly possible to have several graduate students code up vast databases by hand, that map internal linguistic representations to representations (again hand-coded) of objects, states and events in the real world. Using this method, however, means that a database corresponding to (at least part of) the real world must also be developed, which corresponds to the semantic needs of the linguistic representation database. Worse than this task, however, is the task of then mapping the "real world database" into the REAL real world. These problems can be summarised in the question:

> How would I build a natural language processing system to interact with the real world using established techniques?

It is possible for skilled humans to carry on creating databases of representations and semantics until they are blue in the face: there will still not be any easy, natural way to map language to interaction with the real world, even for those with very blue faces.

The problem described above is really a disagreement about the best way to "ground representations" – that is, to provide representations of language with a non-trivial, useful connection to (representations of) aspects of a real environment. This problem has also been characterised as the Symbol Grounding Problem (Harnad 1990), although in this paper we will shy away from referring to representations as symbols as much as possible, since resulting philosophical meta-discussion might obscure the important ideas presented here.

The only useful example available to researchers of a system which can (and does) form non-trivial connections of the kind detailed above is the human being. Rather than applying logic, model theory, graduate students, and the other weighty apparatus of traditional AI research, we have tried to attack this problem in as naïve a way as possible. This is not as ill-considered as one might suppose: naïve approaches to difficult situations are no less likely to result in success than sophisticated ones *a priori*. Further, the naïve approach detailed below has not been possible until recent advances in the technology of artificial neural networks, and so is very much under-explored. It will also be argued later that the methods detailed here, although simple, are capable of being incorporated into larger scale, hybrid systems which require their components to function like traditional computers, in the sense that they must produce (at their interfaces) only discrete well-defined symbols.

At an early stage of their development, humans ground internal representations of language to do with concrete objects in a relatively simple associationistic way. They seem to learn unsophisticated mappings between linguistic stimuli and other sensory stimuli; the exact story varies, depending on whether your beliefs tend towards Behaviourism or Interactionism. A similar approach using machines might prove to be an appropriate solution to the part of the representation grounding problem that deals with concrete concepts. This paper will also show a rather more subtle grounding process: that of associating actions with verbs, rather than names with objects. This is *prima facie* a much harder task, but it will be shown here that our naïve approach can begin to deal with it in a satisfying way.

Several workers in Artificial Neural Networks – (Cottrell et al. 1990, Plunkett et al. 1992, Regier 1992) – in more traditional Artificial Intelligence – (Pustejovsky 1988, Siskind 1990) – and in hybrid systems using both technologies – (Nenov 1991) – have approached the problem of grounding language in sensory stimuli. They have all been successful in some way, but none of them has yet achieved practical applicability in the same way that we think our approach could be said to be practically applicable: it learns in seconds, operates in seconds, and works with real video input, and a variety of linguistic encodings.

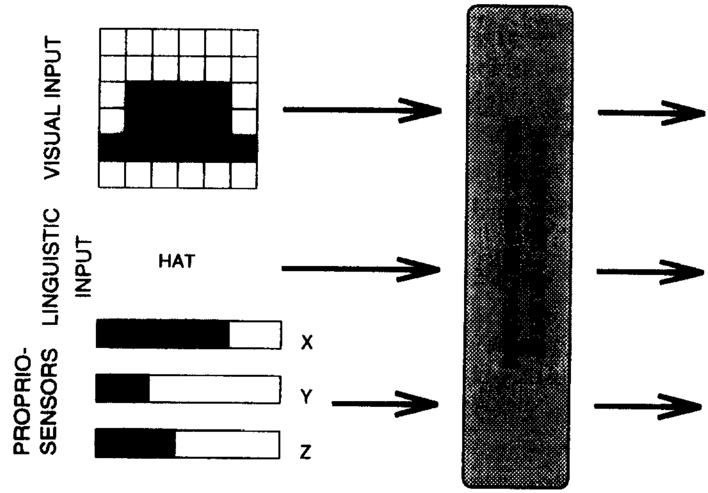

Fig. 1. Schematic diagram showing organisation of "naïve" processing.

3. The naïve approach

In fact, because of the emotive weight of the word, we are probably selling ourselves very short if we refer to our approach as naïve. We will describe its methodology in more detail, and pinpoint the aspects which are relatively "naïve" in comparison with the usual assumptions about natural language processing and representation grounding.

As noted above, humans are an excellent prototype to attempt to emulate in their acquisition of grounded representations: unfortunately, it is physically impossible to emulate their operation in any detail. There is no need to do this to take advantage of their methods, however: simply use a system which can integrate data from arbitrary sources – say sensorimotor, propriosensor, and (read) language – and process them together, without requiring major subsystems to pre-process each modality before being combined into a unified format for central processing, and then possibly translated back into original modalities. While this may be what the brain does in detail – some would say so (Fodor 1975) – at least to the naïve observer, what it does is simply to process any inputs in a highly integrated way, using little in the way of pre-processing. To that end, we have designed a system which processes data from arbitrary sources concurrently, with very little pre-processing, before providing output in all the required formats, using weightless artificial neural networks. The conceptual organisation of the network is shown in Fig. 1.

All inputs to the system are processed in the same network, with virtually no pre-processing. There is no attempt to normalise representational formats:

the network gets crude information, straight from the world. Further, because of the detailed structure of the networks to be used (they are "recurrent"; they contain feedback loops), the "naïve" approach is actually more sophisticated than most other previous approaches.

The use of recurrency, together with weightless artificial neural networks, allows a novel kind of analysis: the systems described later can be understood as special kinds of state machines – Neural State Machines (NSMs) – which have both the functional properties of neural networks, and the computational properties of automata. Thus, although they provide "soft computing" to a user, they can also provide "hard computing" at an interface with more traditional computing methods.

4. Background to the current approach

4.1 *Weightless artificial neurons*

There is an extensive literature of both traditional *artificial neural networks*, and *weightless artificial neural networks*. For works concerning weightless artificial neural networks, see for example Aleksander and Morton (1993), Aleksander (1990). For the sake of clarity, however, a brief overview will be given here, since some aspects of this approach depend on the use of weightless artificial neural networks.

The most commonly encountered artificial neural network neuron model encountered is the "function-of-sum-of-weighted-inputs". This node simplifies several aspects of generic biological neurons into a tractable mathematical entity: it contains afferent (continuous valued) inputs signals – synapses on the dendritic tree – which are multiplicatively weighted, integrated, and passed through a *transfer function* to produce a (continuous valued) output, representing instantaneous spiking frequency at the principal efferent axon. Transfer functions are usually non-linear, continuous and differentiable.

Networks constructed using such models behave as function extrapolators and classifiers, by dividing input space up into discrete convex regions. They achieve computational universality by using a multi-layered structure. This in turn necessitates the use of computationally intensive and uncomfortably slow training algorithms for the networks, such as *error back-propagation* (Rumelhart et al. 1986). We say "uncomfortably slow" because, although such algorithms work, they require of the order of hours or days to complete moderately sized tasks.

Weightless artificial neurons are very different in construction: they use no multiplicative weighing of inputs, (hence "weightless"), and are usually applied to binary input values. They are not limited to the processing of binary values, however: see Clarkson, Gorse and Taylor's work on pRAMs for instance (Clarkson et al. 1989), which allows continuous input values to be processed.

The simplest forms of weightless nodes just store the required output for a given input, as a lookup table, using the input as the index to the table. This means that supervised training of weightless networks is extremely fast: the storage of patterns is analogous to the storage of binary data in Random Access memory (RAM). Weightless nodes are not pure RAM, however. They differ in at least one way, and usually in two ways for advanced use.

Weightless artificial neural networks are usually used in conjunction with *n-tuple* sampling techniques. These have been used to great effect since the 1950s: Bledsoe and Browning wrote a seminal paper on handwritten character recognition using n-tuple sampling devices (Bledsoe & Browning 1959). N-tuple sampling is usually applied to binary data, and involves randomly sub-sampling vectors of (n) inputs from a larger (*k*n) input vector. An output response for each tuple is calculated – often by counting the number of input bits set to "1". By concatenating the outputs of all samples, a fingerprint for each class in input data – in Bledsoe and Browning's case, for each letter – can be generated. New data can then be used to generate a new fingerprint, and some kind of generalisation can be applied, to find the class membership (if any) of the new input data. Such techniques produce inherent robustness to noise, and exponentially reduce the storage cost of the devices, compared with the storage cost for full coverage of a given input vector, compared with the cost of feeding it directly into RAM.

More advanced weightless nodes can be used, in which probabilistic values are stored at input-addressed locations, rather than simple deterministic output values. These are particularly important when n-tuple sampling might produce the pathological case of more than one input pattern, with different required output values, addressing the same location in the node's memory. The use of probabilistic stored values improves generalisation, and allows large networks to be built while preserving good performance.

The neuron model used in the simulations reported here is the General Neural Unit, or GNU. It is the most sophisticated neuron model in the weightless artificial neural networks family, each node having feed-forward and feedback connections, and its own sample of input. It is a probabilistic node model: in this paper, the probability values stored in the neurons are 100%, 50%, and 0% probability of firing. Myers (1989) has studied the use of larger probability alphabets in detail.

Another important aspect of the GNU is that its generalisation is explicitly controlled: unlike other neural network models, generalisation can be controlled by the user, as well as by the implicit control produced by the structure of the data which is present in most models. Generalisation is controlled through the use of a generalisation algorithm, applied either at run-time, in the case of larger networks such as those described here, or at training time, in the case of smaller networks. The generalisation algorithm used here is the standard Hamming distance-based *spreading* algorithm (Aleksander 1990). Using this algorithm, generalisation can be restricted to network inputs which match previously trained patterns to within a certain Hamming distance, allowing more precise operation of the network.

4.2 *The Neural State Machine and dynamic networks*

The overwhelming majority of work carried out with artificial neural networks uses feed-forward network topologies: these are linear structures which can only perform instantaneous classification and estimation. They necessarily can have no state structure associated with their operation because they are not dynamic systems: they map single inputs to single outputs without regard for previous operations.

Tasks which could be categorised as even remotely "cognitive" in nature are time and context dependent. This is particularly true of the ultimate desire of robotics engineers: communications between humans and machines. It is obviously desirable to incorporate feedback into systems which perform *any* time and context dependent tasks. It is possible to do so in principle for any feed-forward (i.e. non-feedback) neural network; in practice, however, adding feedback often presents difficulties in training, because the complexity of the task to be learned increases, forcing the use of training algorithms of even greater sophistication and time-consumption.

The Neural State Machine, proposed in Aleksander and Morton (1993), is an alternative to these networks. It provides a dynamic system with a state structure, while still admitting the major benefits of weightless node models over other models: their fast training time, and relative ease of analysis. The particular NSM which is simulated in most of the experiments described below has both "internal" and "external" feedback pathways: internal feedback is provided by passing a (possibly under-sampled) version of the state back into the network inputs. This is effectively feedback of visual and linguistic state. External feedback is provided by feedback of internal states, with the addition of input from positional sensors (propriosensors), which can sense their x, y, and z (zoom) coordinates in their environment. This feedback is routed through the environment rather than directly through the network.

The main advantage of the NSM is not that it uses feedback, and so is a dynamic system, however, although this is vital for the particular application. The main advantage is the fact that the neural network can have its states

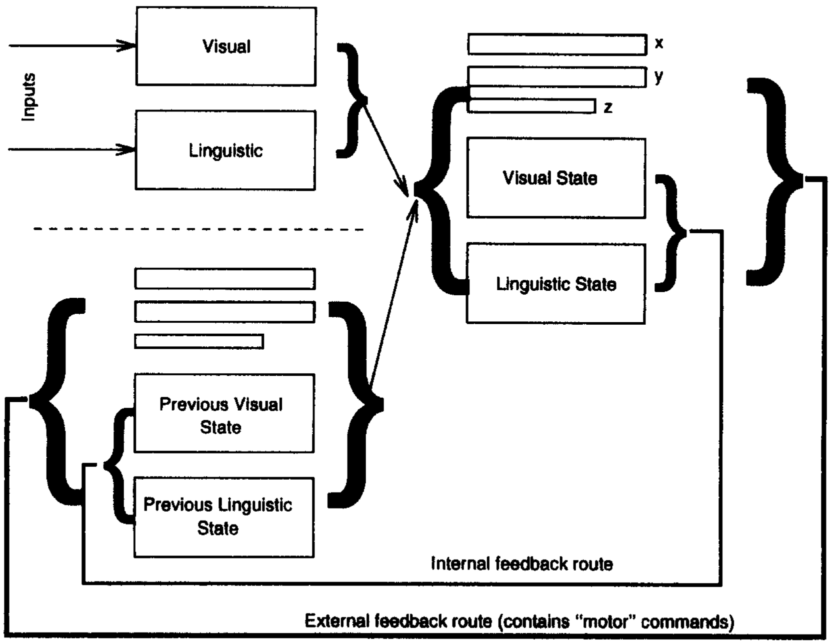

Fig. 2. The Neural State Machine used in the object grounding experiments.

extracted in a reliable way, and so can be viewed, through a suitable interface, as a state machine. Other attempts have been made to analyse traditional artificial neural networks as state machines: Kolen (1994) has shown that these analyses are open to pathological cases. The same is not true of the NSM analysis, since mappings between states and neural activity can be defined unambiguously.

A schematic diagram of the NSM used in the first two sets of experiments is given in Fig. 2. A schematic diagram of the NSM used for verb grounding is given in Fig. 3.

4.3 *Simulation details*

The MAGNUS weightless artificial neural networks simulator (Aleksander et al. 1993, Aleksander 1994) was developed at Brunel University and Imperial College as part of a previous SERC-funded research project. It was used to simulate the large networks with the topology shown in Figs. 2 and 3, from GNU nodes. The network contained approximately 5000 nodes with 32-bit inputs. The experiments reported used input "retina" sizes of 64×64 bits.

The system was operated in KITCHENWORLD, a simplified real-world environment consisting of video camera pictures of objects on a kitchen

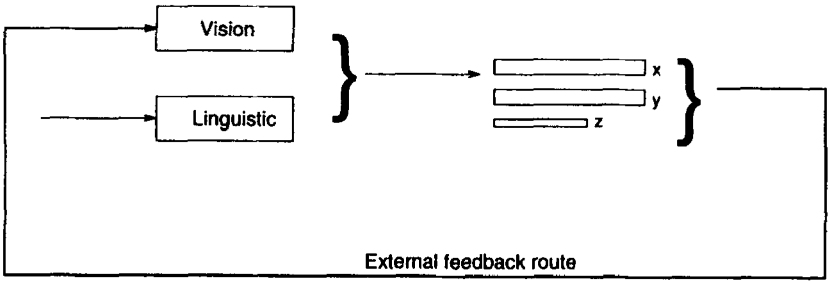

Fig. 3. The Neural State Machine used in verb grounding experiments.

table. There were several examples of each object type – apple, jar, cup – in the environment. The Iconic Training method (Aleksander and Morton 1993) was used to train the network. This involves supervised training of the network into a state with the "iconic" (i.e. sampled) version of the network output as the network's feedback inputs. This method allows the building of attractors in state space without the need for lengthy iterative training. The visual training data for the experiments is shown in Fig. 4. Linguistic training data was in the form of scanned printed words; propriosensory data was in the form of (x, y, z) coordinates.

5. Simulations

In all cases reported below, input to the neural networks consisted of grey-scale video input to the visual modality, and linguistic representations, such as typed words, or symbolic pictures, to the linguistic modality. Video input was not specially pre-processed: it was simply binary thresholded and dithered. Outputs from the networks consisted of a linguistic state, a visual state, both of which were used in feedback to the network, and a set of outputs representing motor commands for movement to a specific position in the KITCHENWORLD environment (expressed in (x, y, z) coordinate form). In the case of simulations investigating verb grounding, output was simply motor commands, demonstrating the semantics of the verb in question. It is possible to dynamically include or exclude motor commands from feedback, which changes the function of the system as described. Performing recall with the network while feeding back the network's perceived (x, y, z) position is termed an *external recall.* This kind of recall causes the system to search around in KITCHENWORLD for its original exemplar for an object classification. When several prototype objects of the same class are trained, the system jumps probabilistically between them when performing external recalls. An *internal recall* is carried out when the only feedback in the system is taken from visual

Fig. 4. Visual training data for the object grounding experiments.

and linguistic states. This causes the system to reconstruct visual and linguistic images on its state outputs without moving around in KITCHENWORLD.

6. Object naming

In the first experiment, object naming, the network was trained to associate the visual input and spatial coordinates of one of the apples in the environment with the input word "apple". Similar training was carried out for a "jar" in the KITCHENWORLD. The system was then shown visual input of other apples and jars in its environment. Using internal recalls, it correctly named them as "apple", or "jar". It named them by recalling an internal state of "apple" or "jar" linguistic input, together with the visual input of the object in question. If the system was allowed to move in its environment, by performing external recalls, it immediately homed in on its prototype object. If the system was shown a novel visual input, it made no erroneous recall when generalisation was restricted to a certain Hamming distance – typically around 50% of the total number of inputs to each neuron. The dynamic control of generalisation is one particularly attractive feature of weightless artificial neural networks, which allows the functionality of networks to be changed, or to change themselves, as they operate. Figures 5 to 7 show the network in operation.

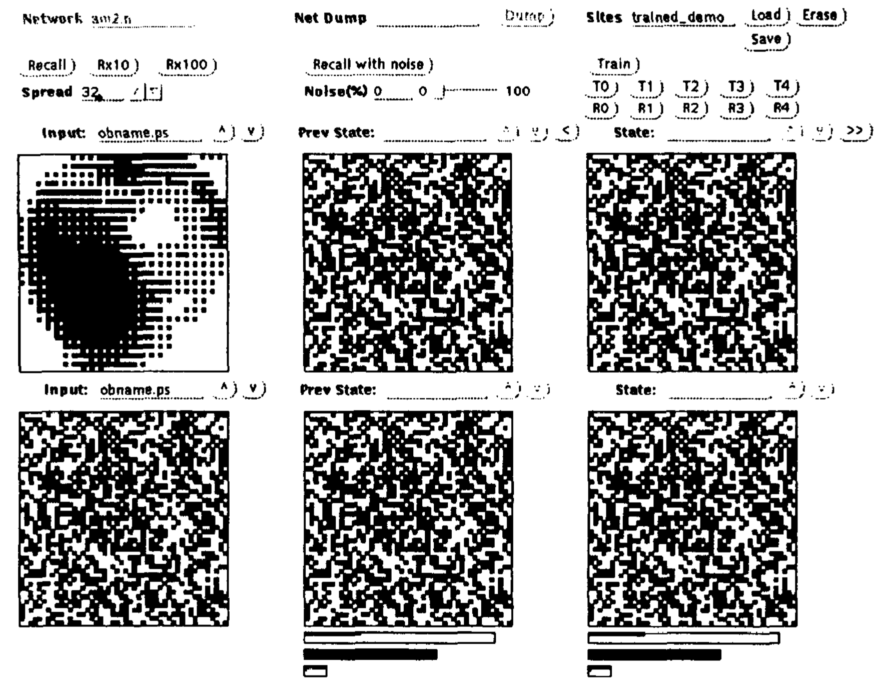

Fig. 5. Recalling the name of the input picture: after initialisation.

This experiment produces correct internal state recall in 2 or 3 steps: one further external recall step is required to locate a prototype object in KITCHENWORLD. This corresponds to about 3 seconds processing time on the testing machine, a Sun SPARCstation 5/70 with 32 MB real memory.

7. Searching for objects by name

For the second experiment, the network was trained as in the first experiment, but was used to search for objects by name. The system was shown the input word "apple" or "jar", and successfully retrieved an internal visual image of the appropriate object. Once again, if allowed to move, by an external recall, the system would home in on a prototype. If the system was shown a novel input word, providing generalisation was restricted, the system did not make an erroneous recall. Figures 8 to 10 show an example run of this experiment.

The two experiments above used only a fraction of the storage capacity of the network. The child language development literature usually treats a lexicon of about 50 words as the standard size for the first stages of lexical development. The system described here can easily acquire 50 grounded

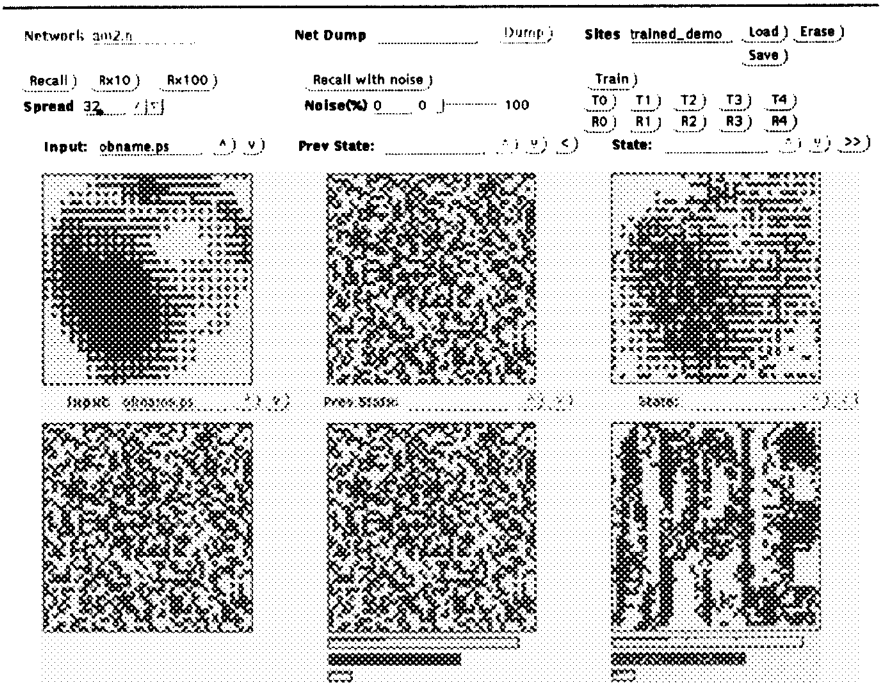

Fig. 6. Recalling the name of the input picture: after one (internal) recall.

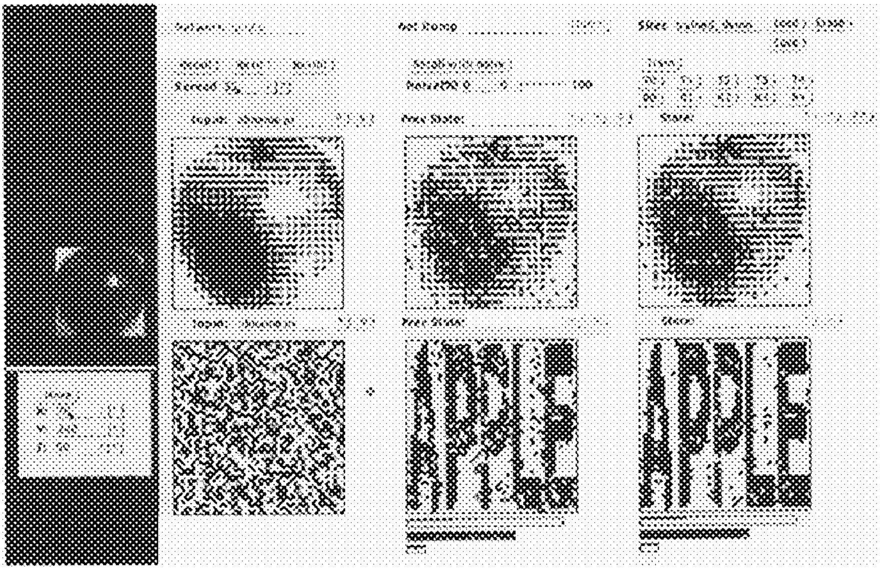

Fig. 7. Recalling the name of the input picture: after two (internal) recalls and one external recall.

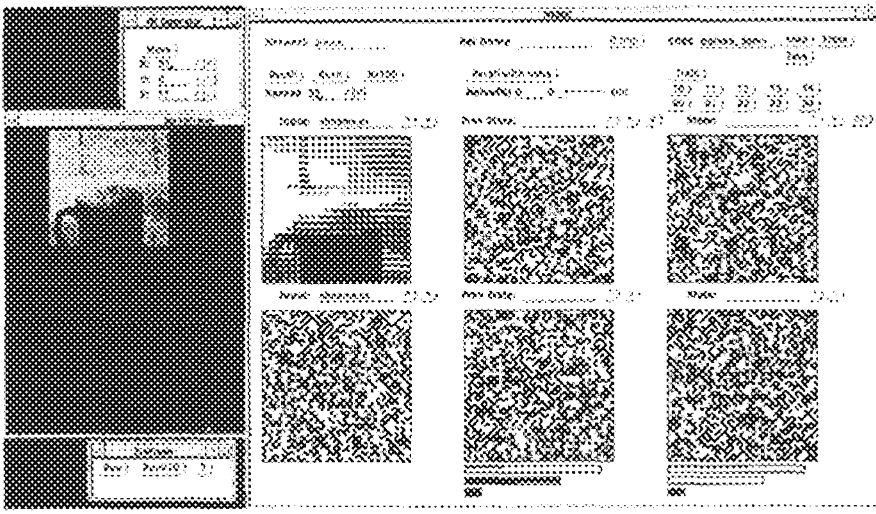

Fig. 8. MAGNUS initialised to a random state after training.

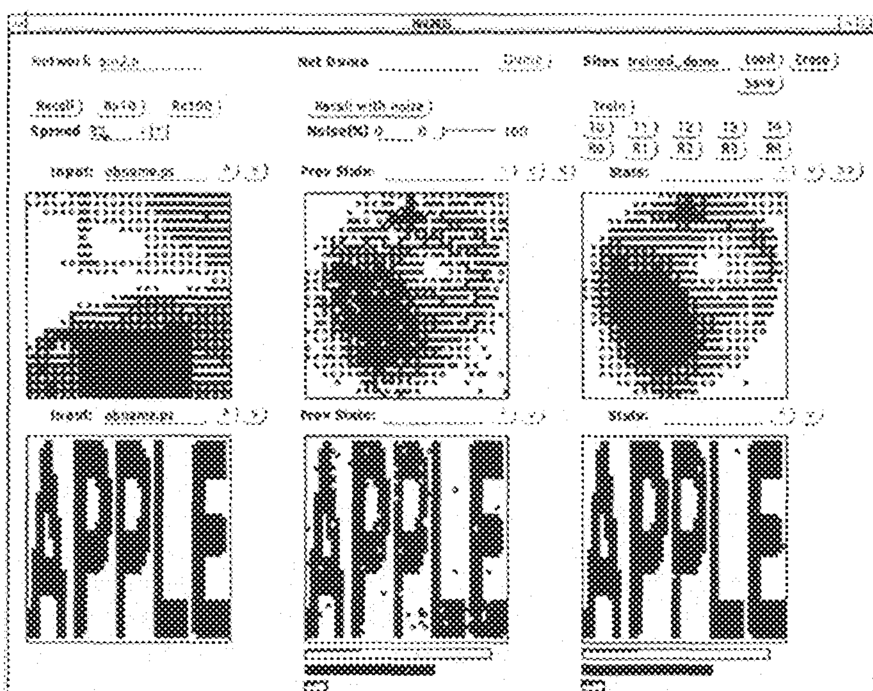

Fig. 9. The system after two (internal) recalls – almost perfect recall.

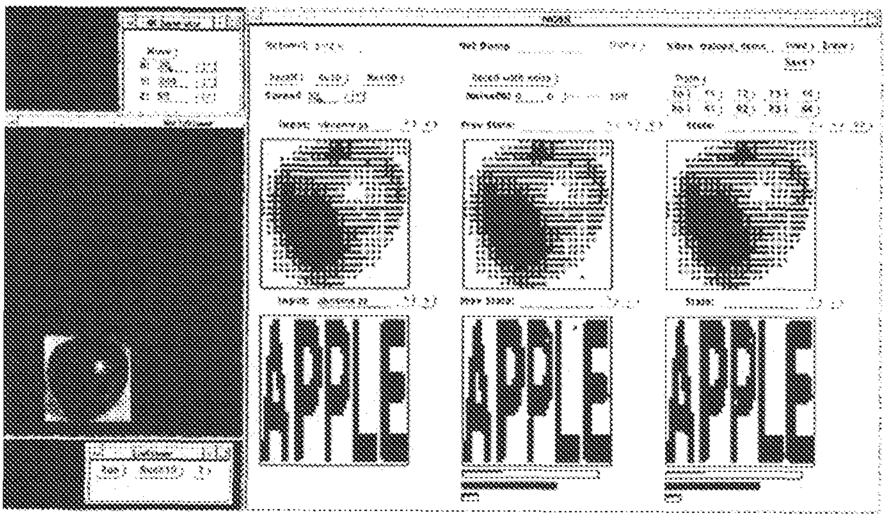

Fig. 10. The system after a third (external) recall: now the retina is in the correct place.

concrete nouns, and can potentially store 2^{32} different inputs. A more practical upper bound for the network described here is a storage capacity of 10000 "patterns" – object-prototype/object-name pairs – assuming that they are reasonably orthogonal in the input space.

The system takes 0.5–1.0 seconds to learn each association of visual and linguistic input, and around the same time to make a single recall step: no more than 10 recall steps are usually required to retrieve correct representations in cases where the network is quite saturated. In cases such as those shown above, where the network is not saturated, only 2 or 3 steps are needed, as noted already. Thus the system can easily be trained and operated in real-time, a distinct advantage over more traditional neural network methods which would typically take several days to learn a training set of this complexity. The MAGNUS simulator allows off-line batch training, which automates the learning of large numbers of word-picture pairs.

8. Action verb grounding: building motion space attractors

Preliminary experimental results have been obtained to demonstrate the properties of a virtual robot – represented by the position of the retina in KITCHENWORLD – whose patterns of movement are modulated by means of a linguistic input consisting of action verbs. Figures 11 and 13 show the experimental simulation in operation. The controlling neural system is configured with external rather than internal feedback and has no access

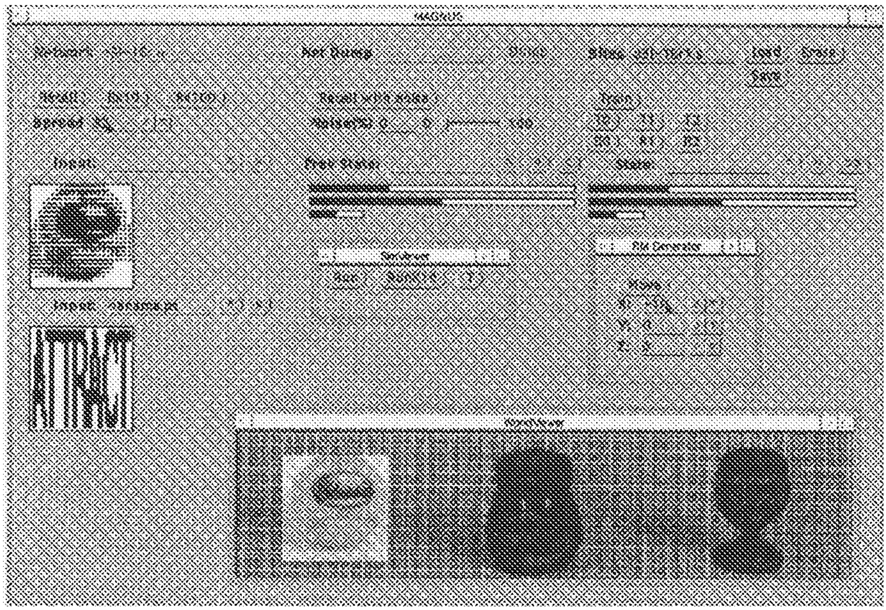

Fig. 11. MAGNUS verb grounding simulation using "attract".

to information regarding the robot's absolute location within the scene. All movement of the retina (robot) in this simulation is relative to its current location and is controlled by the patterns on the output of a set of move neurons. To simplify interpretation of results the robot is constrained to move with one degree of freedom in the X direction. The robot is trained on two conflicting patterns of motion with respect to a target object, in this example the tennis ball. Each activity is associated with a different action verb pattern.

The first motion is towards the visible target object and is generated by three training steps. These are moving the robot to the target from the left side, moving to the target from the right side and finally allowing the robot to remain motionless on the target. All such associations are formed with the "attract" verb present on the linguistic input. The second pattern of motion is trained by moving the robot away from the target form both the objects left and right hand side. This training is achieved with the "repel" verb present on the linguistic inputs.

Generalisation of the five resulting trained patterns is accomplished using the spreading algorithm. This results in a very large probabilistic state structure within the neural system which can be investigated by measuring the moves that are most likely from a given location along the X axis. The experiment consisted of placing twenty robots at every one of 489 locations along

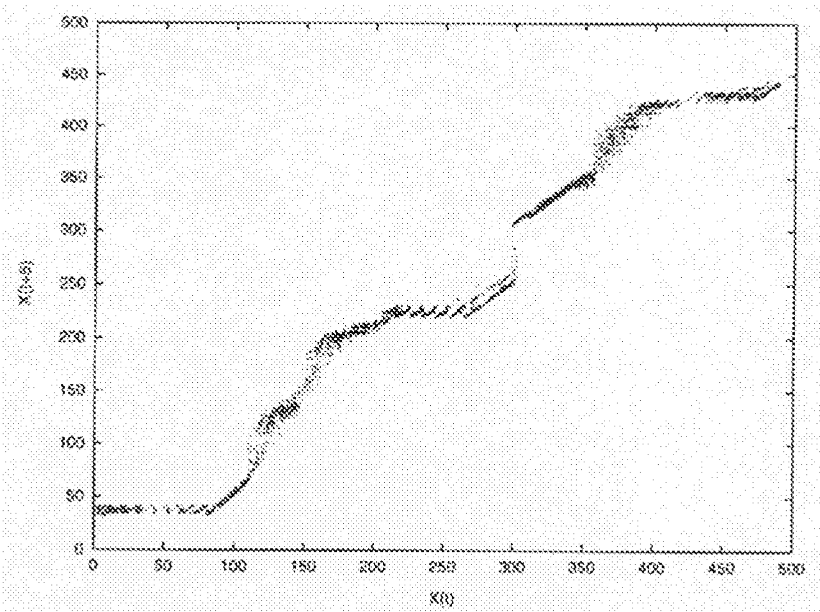

Fig. 12. Dynamics with "attract" verb.

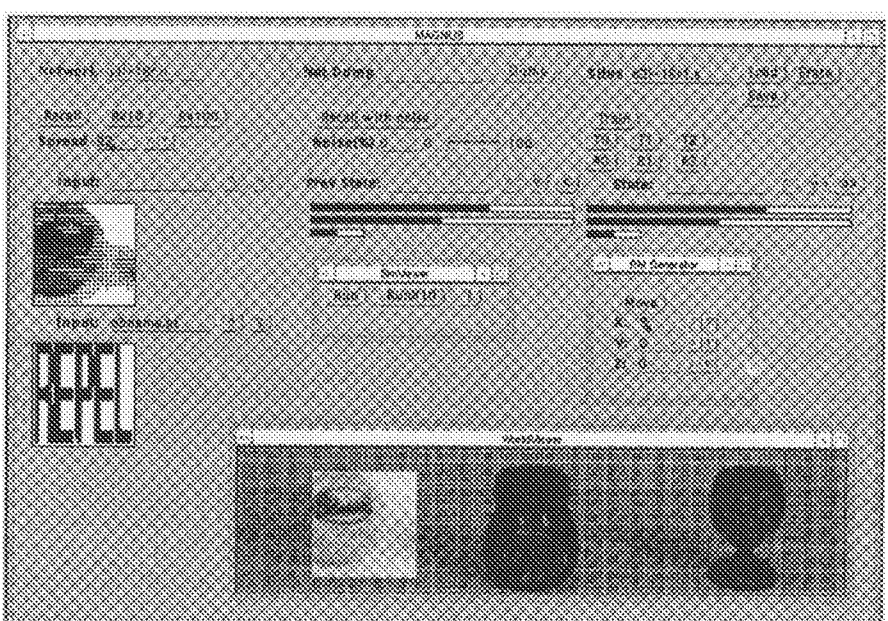

Fig. 13. MAGNUS verb grounding simulation using "repel".

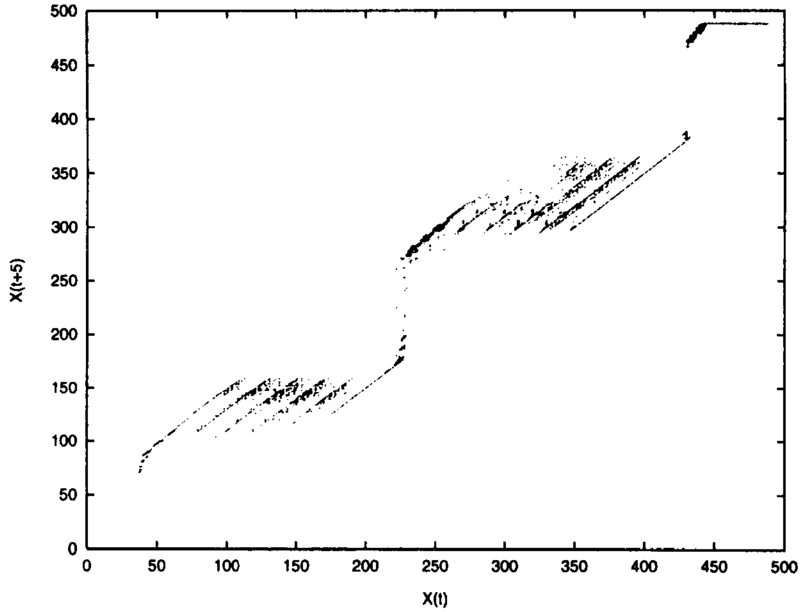

Fig. 14. Dynamics with "repel" verb.

the X axis and allowing each to make a sequence of five moves. This was carried out twice, first with the "attract" pattern and secondly with the "repel" pattern present on the linguistic input. In each case the correlation between each robots X position at time $t = 0$ and $t = 5$ was plotted as a scatter graph (see Figs. 12 and 14). It can be seen that generalisation results in the formation of Motion Space Attractors in the vicinity of all three objects in the scene even though these objects look significantly different to a human observer. The attractors are visible in the scatter plots as the horizontal regions at $X_{t+5} = 38$, $X_{t+5} = 234$ and $X_{t+5} = 430$. Any robot starting close to one of these locations is drawn into an attractor. In the complementary case, the "repel" verb causes robots to be repelled from these locations. This is seen in Fig. 14 where there are vertical regions at the trained locations and clustering of robots in no man's land in between.

9. Discussion and criticisms

The method demonstrated in this paper for noun and movement verb grounding is that of creating active visual-linguistic memory of specific scene elements, to which generalisation and time-dependent sequence processing is applied. It might be thought that this is no more than simple associationism:

this is not a justified criticism, since the system exhibits not only significant levels of generalisation, but also excellent levels of accuracy with novel input nouns or object pictures, when used for noun grounding. The accusation of associationism in the context of verb grounding is clearly not valid, since the system demonstrates powerful generalisation on different objects within the visual field.

The examples shown here were relatively small: other work (Plunkett et al. 1992, Sales 1995) has shown that the storage capacity and performance are easily extended to the 0 (50) classes of words which were mentioned above to be a common assumption in the child language acquisition literature. A further extension of this capacity to a size reasonable for working with human beings in a relatively specific, though not specialised, environment is easily envisaged.

Given suitable robotics, these grounding systems could thus find practical applications in the real world. The most important practical aspects of the system are its speed of training, compared with traditional artificial neural network models, and its applicability to real data, rather than requiring sophisticated pre-processing. The systems described above are practically applicable in this sense precisely because they are naïve, however: the implications of their naïvety must be assessed before we hail them as the solution to all representation grounding problems.

9.1 *Implications of the naïve approach*

The most serious criticism which should be levelled at the methods described above is that the systems shown rely heavily on the form of object in visual input for their performance. This is only partly valid, and even when it carries weight, we must ask whether this is a problem, a feature, or a benefit. It is true that without sophisticated pre-processing, visual input images of cats in different positions, say, will vary tremendously, as will visual input images of different breeds of cat, say, in the same positions.

Remember, first, that this approach is intentionally naïve: it does not seek to acquire a full adult vocabulary and associated concept manipulation apparatus for objects in the everyday world. What it *does* seek to achieve is to show how, in principle, real data can be used quickly and effectively to produce grounded representations of linguistic (and propriosensory) input data. Now that the applicability of our methods to our chosen demonstration problems has been proven, we can apply ourselves to integrating more sophisticated behaviour into the system. It would be possible, for instance, for pre-processing to provide visual image segmentation and normalisation, which would certainly help the representation grounding systems to generalise correctly over large, highly varying classes of exemplars. It must be remembered, however, that even remotely human-like object recognition and categorisation is not achieved without other cognitive processing. The incorporation of grounding systems into larger hybrid systems, which could provide greater sophistication of processing, is discussed below.

Another criticism of the approach above is that the linguistic representations it uses are unrealistic: it would be more useful to use a system which processes phonetic information. Other work has shown that the use of phonetic information is perfectly possible, and may in fact improve system performance, when compared to that obtained using alphabetic linguistic inputs.

10. Conclusion

A deliberately naïve approach to the grounding of linguistic representations in sensory information using weightless artificial neural networks has been demonstrated. Although naïve in the sense that it uses no sophisticated preprocessing of input data and does not make major restrictions to its working environment, it has been shown that useful associationistic connections can be made between linguistic input, propriosensory input, and visual input. The examples shown here have been very simple: in practice the systems demonstrated have been shown to have a large capacity for learning such associations.

Truly useful behaviour by a grounding system, such as the identification of cats of many different breeds in many different positions as "cats" is simultaneously dependent not only upon information derived directly from the environment, but also upon contextual information which has to be borne in an intelligent agent's mind. The integration of more traditional systems which can process and store such contextual information, and representation grounding systems such as those demonstrated in this paper is only possible if the interface between them is adequate. Too often in such hybrid systems, interfacing is hampered by operations like thresholding of continuous values into discrete values. With the benefit of the analysis of weightless artificial neural networks into NSMs, it is possible for both parts of the hybrid system to share the same state-based language, without valuable information being lost in translation.

Thus, Artificial Intelligence systems could have access to an unambiguous state structure representing the operation of the neural state machines: this is in contrast to previous work which has shown that other types of network are prone to pathological cases, making it unsafe to "extract" state structures from their operation for use in traditional computing techniques.

The practicality of such interfacing brings an important advance to the field of natural language processing: the application of artificial neural networks to representation grounding means that non-trivial connections can be established between environments which are situated within the real world, and nouns and verbs which refer to them. This is a first step in solving the so-called "symbol grounding problem", which has always rendered brittle the artificial intelligence methods applied to real environments.

References

Aleksander, I. & Morton, H. (1993). *Neurons and Symbols*. Chapman and Hall: London.

Aleksander, I. (1990). Ideal Neurons for Neural Computers. In *Parallel Processing in Neural Systems and Computers 1990*, pp. 225–228. Elsevier Science Publishers: Amsterdam.

Aleksander, I. (1994). Developments in Artificial Neural Systems: Towards Intentional Computers. *Science Progress* 77(1/2): 43–55.

Aleksander, I., Evans, R. & Penny, W. (1993). MAGNUS: An Iconic Neural State Machine. In Proceedings of *The Weightless Neural Networks Conference*. York: University of York.

Bledsoe, W. W. & Browning, I. (1959). Pattern Recognition and Reading by Machine. In Proceedings of *The Eastern Joint Computer Conference*.

Clarkson, T. G., Gorse, D. & Taylor, J. G. (1989). Hardware Realisable Models of Neural Processing. In Proceedings of *The 1st IEE Conference On Artificial Neural Networks*, pp. 242–246. London: IEE Conference Publications.

Cottrell, G. W., Bartell, B. & Haupt, C. (1990). Grounding Meaning in Perception. In Marburger, H. (ed.) *Proceedings of the 14th German Workshop on AI*, pp. 307–321. Springer Verlag: Berlin.

Fodor, J. A. (1975). *The Language of Thought*. Cromwell: New York.

Harnad, S. (1990). The Symbol Grounding Problem. *Physica D* **42**: 335–346.

Kolen, J. F. (1994). Fool's Gold: Extracting Finite State Machines from Recurrent Network Dynamics. In Cowan, J. D., Tesauro, G., Alspector, J. (eds.) *Advances in Neural Information Processing System 6*, Morgan Kaufman: San Mateo, CA.

Myers, C. E. (1989). Output Functions for Probabilistic Logic Nodes. In Proceedings of *1st IEE International Conference on Artificial Neural Networks*, pp. 163–165. London: IEE.

Nenov, V. I. (1991). *Perceptually Grounded Language Acquisition: A Neural/Procedural Hybrid Model*, Ph.D. thesis, available as Tech. Report: UCLA-AI-91-07, Artificial Intelligence Laboratory, University of California Los Angeles, Los Angeles, CA.

Plunkett, K., Sinha, C., Møller, M. F. & Strandsby, O. (1992). Symbol Grounding or the Emergence of Symbols? Vocabulary Growth in Children and a Connectionist Net. *Connection Science* **4**(3/4): 293–312.

Sales, N. J. (1995). *Aspects of Lexical Acquisition in Artificial Neural Networks*. Ph.D. thesis, Imperial College of Science Technology and Medicine, University of London.

Pustejovsky, J. (1988). Constraints on the Acquisition of Semantic Knowledge. *International Journal of Intelligent Systems* **3**: 247–268.

Regier, T. (1992). *The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization*. Ph.D. thesis, available as Tech. Report: TR-92-062, International Computer Science Institute, Berkeley, CA.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature* **323**: 533–536.

Siskind, J. M. (1990). Acquiring Core Meanings of Words, Represented as Jackendoff-Style Conceptual Structures, from correlated Streams of Linguistic and Non-Linguistic Input. In Proceedings of *The 28th Annual Meeting of the Association for Computational Linguistics*. Pittsburgh, PA: University of Pittsburgh.