Natural Language Parsing using Cell Assemblies: computational linguistics with attractor nets

C. Huyck, R. Belavkin, and C. Connolly
Middlesex University

Introduction

We will develop a software agent based solely on simulated neurons that uses natural language. A parser based on simulated neurons will have several benefits over symbolic parsers and even other types of connectionist parsers.

Computationally, neural processors have an obvious mechanism of symbol grounding; symbols can be grounded in sensory data giving a mechanism for at least basic sensory semantics. So neural processing has a hope of resolving the classic AI symbol grounding problem (Fodor 2000), since its symbols are grounded in sensory inputs. Semantics are critical for resolving parsing ambiguities and for robust parsing. Using grounded symbols as the basis of semantics will allow the parser to be more effective.

Scientifically, neural simulations can model psychological phenomena and can be compared to neural data by, for example, fMRI and electrodes on individual neurons. Consequently, psycholinguistic and neural data can be directly compared to the parsing system.

Theoretically, processing with attractor networks can be advanced. Attractor nets settle into stable or pseudo-stable states as processing progresses. Simulated neural nets may act as attractor nets, and most attractor net applications are limited to categorisation. We will build a parser with the lexicon, grammar rules, and rule application mechanism all encoded in neurons. These active symbols (Kaplan, Weaver, & French 1990) will process and thus will apply rules and build structures. Cell Assemblies (CAs) are a higher order structure within the network of neurons. They are sets of highly connected neurons that remain active even after external stimulation ceases (Hebb 1949). Individual CAs are single attractor states, but applying rules rapidly generates new attractor states, and allows the system to move to new states. This real application will be an advance on current attractor net applications.

A neural system that can understand natural language, that is situated in an environment, and can learn, will be a major advance over any existing system. Indeed, it will be a significant step towards an intelligent system.

Background

Parsing and Related Natural Language Technologies

In the past twenty years, Natural Language Processing (NLP) technologies have moved from academia into the business world. Three major reasons contribute to this move. First, technologies have reached a stage of sufficient maturity that they can work with real world data. Second, NLP has made use of machine learning to develop systems that learn key features of the domain that is being processed. Third, the popularity of the Internet has led to a vast quantity of textual data that needs to be processed in a range of ways so that it can be used (Church & Rau 1995).

Parsing is the central NLP subtask, involving processing a sentence to recognise its syntactic structure, and it may generate semantic structure. A standard mechanism for measuring the effectiveness of a parser is to gauge its performance on a large textual database such as the Penn Tree Bank (PTB) (Marcus, Santorini, & Marcinkiewicz 1993). The best parsers score around 90% on this task. Collins' system is a good example of a high scoring parser (Collins 1997), although this is an entirely syntactic measurement. The best work in this area has taken advantage of machine learning techniques to learn probabilities for selecting between ambiguous syntax trees.

While the NLP field continues to advance, the pace has slowed. This may be due to a limitation of current symbolic techniques. With the parsing problem, systems are on the 90% plateau largely because ambiguity arises from the semantics and pragmatics of the sentence. The semantics of at least concrete nouns are based, to a large degree, on the things that can be done with the referent of the word (Gibson 1986). So *pen* is intimately related to *writing* because that is what one does with a *pen*. Current machine learning techniques are unable to handle this type of natural semantics. Instead they are usually based on a "bag of words" approach where the semantics of a word is based on the frequency of the words with which it appears (Lewis & Ringuette 1994).

More importantly, NLP needs to use semantics because it is the semantics that are of interest. The parsers that perform best on the PTB do not generate a semantic analysis; they are built to solve the syntactic task. There are no standard tests for semantic evaluation, though some tests have been proposed (Gaizauskas, Hepple, & Huyck 1998).

An existing system, Plink (Huyck 1994; 2000), uses semantics and syntax in parallel during parsing. By using this information, preferences can select the appropriate rule to apply, parsing in linear time, to some extent like people. It is important for computers to process language as people do. Unlike formal languages, natural languages are ambiguous, are frequently ungrammatical, and often omit important features. By processing language like humans, the system will be able to use meanings that are similar to those of people. Existing text engineering systems are usually built to solve one task, like extracting database templates for rocket launches (Huyck 1998). They are somewhat successful, but are limited by their singular nature and have to be re-engineered for another task. A system that processed language like a human would be able to function in a wide range of domains. Since Plink parses like humans, it is more closely related to psycholinguistic research than most other parsers.

Connectionist Parsers

There have been some connectionist Natural Language Parsers including those by Mikkulainen (1993) and Henderson (2000). These parsers are not based on anything like real neurons and they have had basic problems such as their inability to accomplish real world tasks. These problems are usually solved by a symbolic solution. Knoblauch's system (Knoblauch, Markert, & Palm 2005) uses a more interesting solution. It is based on model neurons with a medium degree of biological faithfulness, implements a regular grammar, grounds its semantics in the real world, and resolves ambiguity dynamically.

While Knoblauch's is the most sophisticated CA-based system that we are aware of, it still has shortcomings. Firstly, its parsing mechanism is based on Regular Grammars; some support the notion that language is Regular (Blank 1989), but the consensus is that natural language is at least Context Free. Secondly, each word is preallocated neurons that are later connected to sensory input; clearly words are not preallocated particular neurons in the brain. Both of these problems can be avoided, but require greater complexity. Thirdly, the binding is done via synchrony. This has a limit to the number of different bindings, but also requires a great deal of computation because it needs a fine time grain for simulation ($\leq$ 1 ms) (Connolly, Marian, & Reilly 2004).

Computation with Neurally Plausible Attractor Nets

The scientific community has an incomplete understanding of the brain and neurons. However, it is widely agreed that spiking leaky integrators are a simple, largely accurate model of neurons, and that the brain contains CAs (Hebb 1949). These two hypotheses are related and are linked by the theory of attractor nets (Amit 1989).

One large question in neural and connectionist research is what model to use. There are a wide range of models from Hopfield nets to complex conductance based models. Biological faithfulness usually increases simulation time, so it is hard to simulate a large number of very sophisticated neurons. An example of the time vs. faithfulness tradeoff is how frequently each element is updated. Neural models that are updated every 1 ms of simulated time usually incoporates spike transmission times and refractory periods. Models with larger time grains do not need to account for these phenomena.

Though incomplete, spiking leaky integrators are a good model for neurons because they are a simple model that is inexpensive to simulate. They also have key features of neurons: 1) leaky integrators collect activation from their inputs; neurons collect activation from other neurons; 2) when an integrator collects enough activation, it fires; when a neuron has enough activation it sends spikes to connected neurons; 3) if the integrator does not fire, some of its activation leaks away; if a neuron does not fire some of its activation leaks away.

A network of leaky integrators may act as an attractor network. If enough integrators fire and their mutual connections are strong enough, they will enter an attractor state where a large number of the integrators are firing and will continue to fire. If enough neurons spike and are connected with sufficient synaptic strength, they will cause a CA to ignite. According to Hebb (Hebb 1949), CAs are the neural correlate of long-term human concepts, and an active CA has been correlated with buffers, short-term memory and other types of working memory (Hebb 1949; Kaplan, Weaver, & French 1990; von der Malsburg 1986).

Neural nets and connectionist systems in general are almost always used for categorisation tasks. Attractor nets go to a state and stay there. However, animals cannot have their brains in a single state; they must move on to new states. Certainly environmental input has an effect on the stability of a state, but attractor nets can be modified to move on to new states in a principled way.

To bridge the gap to human level cognition, other types of calculations need to be implemented. For example, it has been proposed that rules can be implemented with attractor nets (Huyck 2001b).

While other systems can implement rules, calculating with attractor nets is of particular interest because they can learn. Work in learning with attractor nets is largely based on learning categories. There has been little work on learning other processes and since there is little work in rule-application, there has been no work in learning rules in attractor nets.

Connectionist systems also make use of parallelism. So attractor nets that implement complex calculations on a parallel machine may be very fast.

Our Work to Date

The PI's background is in NLP. He developed a symbolic human-like parsing system, Plink (Huyck 1994; 2000), that parses linearly and robustly. It is based on a large number of preferences that determine the parse to pursue; a small look-ahead allows Plink to avoid backtracking. It uses a unification-based grammar, integrating syntactic and semantic processing. Plink performs well on the standard syntax parsing metrics but has not exploited machine learning techniques. Plink has been used in a range of tasks including the DARPA sponsored Message Understanding Competitions (Lytinen *et al.* 1992; 1993; Huyck 1998) and text mining tasks. Plink is also being used in a symbolic conversational games agent (Kenny & Huyck 2005).

It would be relatively simple to use machine learning techniques to set Plink's semantic parsing preferences. However, we were interested in the mechanism that people use, feeling that it would be easy to duplicate other's CA work. To our surprise, we found that very little simulation work had been done with CAs and even less work with CAs for practical AI tasks. Consequently, we have spent the last several years developing programs to solve basic CA problems and some real-world tasks. This work is more fully explained in the Previous Track Record section.

One key weakness of AI connectionist systems is binding (Fodor 2000). Binding enables calculations beyond categorisation, and is easily done on vonNeuman systems. The standard solution to the problem in connectionist systems is to bind via synchronous firing (von der Malsburg 1986), and simulations of this exist (Sougné 2001). We have implemented another type of binding using spontaneous neural activation. Spontaneous neural activation has several interesting computational properties including the spread of activation beyond neurons that are directly activated by external stimulation (Huyck & Bowles 2004). Our simulations (Huyck 2005) using a single layer model have shown that it can enable binding via a form of medium-term memory that is rapidly formed (on the order of seconds), and persists for the order of minutes. This binding via medium-term memory has no limit to the number of different bindings. It is also more computationally efficient than binding via synchrony because it can work with a coarse time grain of 10 ms.

Almost all NLP systems use rules, and grammatical theory is largely based on them. Connectionist systems can apply rules if they have variable binding. So, we now have the capability of developing a neural-CA based NLP system. It is now time to combine our NLP and CA research.

Proposed Project

The ultimate goal of the project is to build a system that is situated in an environment, senses, acts, and uses natural language. It will be based solely on simulated neurons, and will learn critical aspects of the environment.

As this is uncharted territory, an iterative approach will be pursued developing three versions of the system with each scheduled for one year of the project. This will provide steps, periods for reflection, and a good basis for publications throughout the project.

An interleaved engineering and neurobiological modelling approach will be pursued. In the interest of making an advance on a problem, an approach that is not biologically valid may be pursued. However, this "hack" will be replaced at a later stage by a more biologically valid solution. For example, we will initially provide CAs for words with the neurons devoted solely to a given word. Later, we will use a more biologically plausible approach allowing the system to learn the words, and those CAs will share neurons.

Critically, this project will explore the dual dynamics of neural-CA systems. The short-term dynamics of CA ignition and cessation will be explored via computations such as rule application. The long-term dynamics of CA formation via synaptic weight modification will be explored by the formation of medium-term binding CAs, learning rule selection preferences, and rule learning.

NLP tests will use standard measurements such as precision and recall. This will be on a restricted domain, so comparison to general systems will not be readily available. However, comparisons to the symbolic Plink system will be made and the corpus will be made available for others to test their parsing algorithms. The corpora that will be used will be domain dependent text. Initially, these will be sentences that we will generate. These will include ambiguous sentences, complex sentences, and ungrammatical sentences. Later, the text will be used to train the system. Finally, we hope to use the system as an agent in an interactive videogame (see Methodology). This will take open ended text from a user, though it will only properly understand text from the domain of the game.

Methodology

We propose a three step programme with each step encompassing the development of a complete system, test environment, and a test of the system in its environment. All of the systems will be computer simulations of neural processing. The three steps are first to build a parser; second to develop a system that senses its environment and to integrate sensing with parsing; third to enable the system to interact with other agents in the environment while making use of language and sensing from the earlier system.

The first major task will be to engineer a parser. A similar mechanism has already been developed (Knoblauch, Markert, & Palm 2005), but is based on a very simple Regular Grammar, and implements binding via synchrony. This first prototype will implement the Plink algorithm using a unification-based grammar. Additionally, it will bind via short-term synaptic weight change. This first prototype will not learn words.

This will be a significant development, but the initial idea was proposed several years ago (Huyck 2001b). There will be long-term CAs to represent words and grammar rules. Completed and partially completed phrases are represented by medium-term CAs, and these are the stack elements; this medium-term memory implements binding. Both rule application and selection are required. A rule may be applied when the appropriate phrases and words are on the stack. The application of a rule changes the contents of the stack. Though the algorithm was proposed several years ago, an appropriate variable binding mechanism, which is necessary to implement rules, has only recently been developed. Word representation is crucial, but in this step will mimic earlier work (Knoblauch, Markert, & Palm 2005) and use predefined sets of neurons for each word. For any given stack state, several rules may be applicable. The symbolic Plink parsing mechanism uses preferences to resolve these ambiguities; the neural-CA parser will use competition between CAs to implement these preferences. The preferences will be encoded in the weights between rules, words, and larger constituents.

The binding mechanism uses spontaneous neural activation (Huyck & Bowles 2004) and a compensatory learning rule (Huyck 2004) to implement medium-term CAs. A special binding area is set up with slightly different neural parameters and spontaneous activation. This area has synapses to and from the areas with long-term memories. When CAs in these areas are active, they rapidly form a new CA that includes the original long-term CAs and neurons in the binding area. This can be used after the long-term CAs have stopped being active, but is erased relatively quickly via synaptic weight change driven by spontaneous activation.

The test of this system will be its production of correct semantic interpretations in the form of frames (Filmore 1968). The frames will initially be generated by statistical analysis of the firing behaviour of the network after sentence processing. Later, frames will be generated by associating slot labels with concepts via simultaneous presentation. The labels will activate when output is requested by a trained signal. This is a type of actuatator.

The second step is to integrate sensing into the system. This will enable the system to learn words based on the environment. Again earlier work (Knoblauch, Markert, & Palm 2005) has done this but the words were prespecified and were associated with the sensed items. The second prototype will instead use a form of labelling where the system learns the neural representation for the words. In essence, this is a neural implementation of the bipolar representation of words described in Cognitive Grammars (Langacker 1987). This will provide CAs for words and semantically related words will share some neurons. This simulated learning of words is unique.

The environment will be the virtual environment of an interactive videogame. We have some experience in this domain (Kenny & Huyck 2005), it is an industrially important domain (Laird & VanLent 2001), and the videogame environment is very flexible. This flexibility means it can range from a relatively simple domain of simple objects to a very sophisticated dynamic one with active agents. The environment will be the Sierra Half-Life (Sierra Inc. 2005) system; this is flexible shareware game technology that has 3-D graphics capabilities, internet capabilities, the ability to create new environments, and hooks for agent development.

The system will be able to resolve syntactic parsing ambiguities like PP-attachment ambiguities. Ambiguity resolution will be significantly based on the words.

Testing of the second prototype will repeat the semantic test from the first step. The system should also be able to name concepts in the environment. Finally, ambiguity resolution will be tested using a Parseval like measurement.

The third step will be to build an agent in the Half-Life environment. This prototype will have the ability to move, take other actions and have goals, and these will inform sensing. A spreading activation network (Maes 1989) will be used for a simple form of planning. The network selects actions based on the spreading of activation between competence modules. The spreading activation net should be directly translatable into a neural-CA architecture.

Rule learning will be integrated into this agent. Rule learning is common in cognitive architectures such as ACT–R (Anderson & Lebiere 1998) that are largely symbolic. The system will be able to learn new grammar rules to handle common ungrammatical phenomena. It will also be able to learn behavioural rules to improve its performance. Ideally, it will be able to explore the environment and learn new causal relations.

The SOAR (Newell 1990) and ACT–R (Anderson & Lebiere 1998) cognitive architectures have implemented and tested theories of production rules learning. While SOAR has been using an elegant and powerful chunking mechanism, the ACT–R theory has taken a more constrained, but more controllable approach in which procedural knowledge is formed from declarative representations as a result of generalisation and abstraction (and in some cases specialisation). This mechanism, called *production compilation*, has been extensively used to model many cognitive tasks including language acquisition and use. This project will find a neural equivalent of such mechanism, and the variable binding mechanism using spontaneous activation looks very promising.

A working agent that senses and uses natural language gives us a host of opportunities for exploration. These include, but are not limited to, rule learning, simple conversational output, 2-D mapping, cognitive mapping, language learning, planning, goal mechanisms, learning from instruction, and even problem solving.

The initial tests for this third prototype will be its ability to understand and follow instructions. This will be similar to our evaluation of a symbolic agent in the same domain (Kenny & Huyck 2005). Measurement of the agent's ability to take instructions will include specific tests and open domain tests with users. Rule learning will be tested initially by the ability to generate addition tables. Other tests of more sophisticated tasks, such as mapping and simple conversational output, will be developed as necessary.

Research Impact

The basic engineering tasks will be successful, and this will be a solid advance in computing with attractor nets, with significant ramifications for anyone working in this area. This basic engineering is mostly in the first prototype, but will also involve duplicating sensing research in a neural-CA net, and the development of a spreading activation net.

There will also be an advance in understanding neural-CA dynamics. This includes the interplay between short and long-term dynamics, and the simultaneous use of neurons as transducers, stores, and processors of information. This is largely done in the second phase, though components are in both other phases. In particular, the dual dynamics and the interplay between storage and transduction is essential to the development of natural semantics based on sensing. There will be a significant impact on the learning of concepts, labelling concepts, and the understanding of NLP.

Learning complex relations based on synaptic weight change will be concentrated in the third phase. The development of a system that learns rules based on synaptic weight change will be a major advancement in itself.

The neural-CA development will improve understanding of neural processing and neural learning. The UKCRC sponsored grand challenge on the Architecture of the Mind and Brain and the challenge on Journeys in Non-Classical Computation will also both benefit.

NLP research will benefit from an example system that uses grounded symbols. In a related fashion, parsing theory will be advanced by a novel neural parser.

A games agent that is instructed by natural language and learns as it plays will benefit videogame users. This could also be a new market for videogame companies.

Perhaps most importantly, a situated agent that functions and learns from its environment is an opportunity for a revolutionary advance. The system may be able to learn the ramifications of its actions. This could lead to a system with sophisticated natural semantics based on sensing and action. Additionally, the system could learn new processes. It might be able to modify its spreading activation net so that it can set new goals and have new ways of achieving them. The system may even be able to learn from instruction. It is not currently clear how this work will advance during this grant, but at a minimum, this will be a platform to explore these issues. It is conceivable that this work will lead to a revolution in computer development and cognitive science research based around neural-CA nets that learn; this revolution may be underway by the end of the decade.

Dissemination

We plan on disseminating the results of this work in a range of conferences and journals. We intend to run a workshop on programming with attractor nets at a conference such as the European AI Conference in 2008. Other conferences include the Computational Neuroscience Conference, International Joint Conference on Neural Networks, Neural Computation and Psychology Workshop, the International Joint Conference on AI, the European AI Conference, the International Conference on Cognitive Modelling and the Association for Computational Linguistics Conference. Journals include Neural Networks, Neural Computation, Connection Science, Neurocomputing, Computational Linguistics, and Artificial Intelligence.

A project web site will be maintained. All software, benchmarks, and reports will be made available on the site.

Resource Management

The PI will be responsible for the overall and day to day management of the project. Dr Connolly will work on this project full time and consequently will be responsible for most of the programming. It is envisioned that the PhD student will concentrate on sensing. All three will meet on a weekly basis to discuss advancements and issues. There will be further interactions, including the co-investigator, through writing, system development, and presentations.

This is a major development project in the area of neural computation. It is rare to find experienced researchers with skills in the development of systems based on simulated neurons. The availability of Dr Connolly, a researcher with these skills, is an excellent opportunity for a significant advance. Simultaneous training of a new researcher in this area, the PhD student, should help the project, and will add another, rare, experienced neuro-computational engineer and researcher to the small existing pool.

Dr Belavkin will focus on rules and rule learning. Rules will be implemented in the first phase and rule learning in the third phase. This enables independent streams of development throughout the project. Dr Belavkin's expertise in symbolic rule learning (Belavkin 2005) and cognitive architectures (Belavkin 2001; Belavkin & Ritter 2004) will be crucial to developing mechanisms for learning rules.

A substantial number of presentations will be given during this project to disseminate results in this novel area. Two laptops will be needed for presentations along with a travel budget.

Middlesex University will provide accommodation and use of laboratory facilities for the named researcher and the PhD student. A considerable amount of simulation activity will be done with the University providing computers. It will also supply technical and administrative support for the project.

References

Amit, D. 1989. *Modelling Brain Function: The world of attractor neural networks.* Cambridge University Press.

Anderson, J., and Lebiere, C. 1998. *The Atomic Components of Thought.* Lawrence Erlbaum.

Blank, G. 1989. A finite and real-time processor for natural language. *Communications of the ACM* 32:10:1174–1189.

Church, K., and Rau, L. 1995. Commercial application of natural language processing. *Communications of the ACM* 38:11:71–80.

Collins, M. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the ACL*, 16–23.

Filmore, C. 1968. The case for case? In Back, E., and Harms, R., eds., *Universals in Linguistic Theory.* Holt, Rinheart and Winston Inc.

Fodor, J. 2000. *The Mind Doesn't Work That Way: the Scope and Limits of Computational Psychology.* MIT Press.

Gibson, J. 1986. *The Ecological Approach to Visual Perception.* Lawrence Erlbaum.

Hebb, D. O. 1949. *The Organization of Behavior.* J. Wiley & Sons.

Henderson, J. 2000. Segmenting state into entities and its implication for learning. In Wermter, S.; Austin, J.; and Willshaw, D., eds., *EmerNet: Third International Workshop on Current Computational Architectures Integrating Neural Networks and Neuroscience.* 161–76.

Kaplan, S.; Weaver, M.; and French, R. 1990. Active symbols and internal models: Towards a cognitive connectionism. *AI and Society* 4:51–71.

Knoblauch, A.; Markert, H.; and Palm, G. 2005. An associative model of cortical language and action processing. In *Modelling language, cognition and action*, volume 16, 79–86. World Scientific.

Laird, J., and VanLent, M. 2001. Human-level AI's killer application: Interactive computer games. *AI Magazine* 22:2:15–25.

Langacker, R. 1987. *Foundations of Cognitive Grammar. Vol. 1.* Stanford University Press.

Lewis, D., and Ringuette, M. 1994. A comparison of two learning algorithms for text categorisation. In *Proceedings of the SDAIR-94: 3rd Annual Symposium on Document Analysis and Information Retrieval.*

Maes, P. 1989. How to do the right thing. *Connection Science.*

Marcus, M.; Santorini, B.; and Marcinkiewicz, M. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* 19:2:313–330.

Mikkulainen, R. 1993. *Subsymbolic Natural Language Processing.* MIT Press.

Newell, A. 1990. *Unified theories of cognition.* Harvard University Press.

Sierra Inc. 2005. Sierra half-life.

Sougné, J. 2001. Binding and multiple instantiation in a distributed network of spiking neurons. *Connection Science* 13:99–126.

von der Malsburg, C. 1986. Am I thinking assemblies? In Palm, G., and Aersten, A., eds., *Brain Theory.* 161–76.

Previous Track Record

Principal Investigator:

Dr Christian R. Huyck has been working in AI for over 20 years. He has over 40 publications in AI journals and conferences. His two main research tracks are NLP, and neural-CA nets. He has been a lecturer at Middlesex University since 1998 and is currently a principal lecturer. He heads Middlesex's AI research group consisting of a dozen academic staff and a dozen PhD students.

Dr Huyck received his PhD from the University of Michigan in 1994. While there he worked on a range of AI projects, but concentrated on NLP. He participated in the fourth and fifth Message Understanding competitions (Lytinen *et al.* 1992; 1993) while at Michigan, and subsequently on the seventh (Huyck 1998). He also began collaboration with the interdisciplinary SESAME group while at Michigan. This is a broad group of researchers including researchers from AI, developmental psychology, cognitive psychology, and even architecture. The group is interested in how people think and is informed by a Hebbian neural hypothesis. It will provide an excellent forum for discussion of issues related to this research project.

Dr Huyck's PhD thesis was based on a symbolic parser that was effective as an engineering tool, but also parsed like humans (Huyck 1994). Work on the Plink symbolic parser culminated with an evaluation on the standard parsing metric (Huyck 2000). This evaluation showed that Plink performed well on the standard syntactic measurement, 71/74 Precision/Recall, but below the best existing systems, 88/89 (Collins 1997). To improve the system, some sort of automated technique was needed to learn the parsing relationships. Consequently, Huyck started research into CAs to resolve these problems. Huyck has continued his NLP work (LeThan, Abeysinghe, & Huyck 2004) and is collaborating with John Deere Inc. on Text Mining. Huyck is currently working on a conversational games agent (Kenny & Huyck 2005) that has also given us experience in the games environment.

Huyck has concentrated on CA research for several years proposing it as a good basis for a cognitive model (Huyck 2001a). This work has been based on simple spiking fatiguing leaky integrators that are simulated in discrete cycles that are approximately 10 ms of biological time. These largely adhere to known neural properties with the obvious exception of discrete versus continuous time. The main advantage of these discrete cycles is that we can ignore refractory times, synaptic delay, and transmission times across the synaptic cleft as they are much quicker than 10 ms. Consequently, a large number of neurons can be simulated on a PC in real time.

This platform has been used to simulate a range of behaviours. Like other connectionist systems, it has done real categorisation tasks including information retrieval (Huyck & Orengo 2005), categorisation of ambiguous data (Huyck 2004), and hierarchical categories (Huyck submitted). Crucially, this involves neurons that are in multiple CAs. This platform has been used for a novel form of variable binding via medium-term memory (Huyck 2005).

Co-investigator: Dr Roman Belavkin has been working in Cognitive Science for seven years and is currently a Senior Lecturer at Middlesex. He has focused on Cognitive Architectures, cognitive models of learning and decision–making. Recent work (Belavkin 2005) has studied the effects of entropy in subsymbolic information processing on learning in hybrid architectures and how it affects behaviour (Belavkin & Ritter 2004). Belavkin has also shown how rules contribute to the U-effect in learning (Belavkin 2001).

Roman Belavkin has worked as a senior lecturer at Middlesex since December 2002. He was awarded his PhD in 2003 from the University of Nottingham. His thesis was a multidisciplinary project on cognitive modelling, and it was a candidate for the Distinguished Dissertation 2003. Previously, Belavkin held the position of a research assistant at the Department of Physics in Moscow State University, from which he obtained his MSc degree in atmospheric physics.

Since 1998, his interests have been in AI and cognitive modelling using both symbolic and subsymbolic approaches. One of the main contributions of his work recently has been modelling the effects of emotion on problem solving and decision making. One of the outcomes of this work was the creation of a new subsymbolic learning algorithm for the ACT–R cognitive architecture (Belavkin & Ritter 2004), which is currently used in studies of the effects of caffeine, stress and other factors on cognitive processing in humans.

Belavkin is also interested in neural and Bayesian learning algorithms. He is a member of the EPSRC funded research network on Independent Component Analysis as well as the Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB). At Middlesex, Belavkin is a member of the AI research group and the Interaction Design Centre.

Named Resarcher:

Dr Colm Connolly has recently received his PhD from University College Dublin. His thesis was based on a model of the visual system. This model was divided into the "what" and "where" pathways characteristic of mammalian visual systems. It was constructed from laterally connected excitatory and inhibitory spiking neurons which were trained with the SOM learning algorithm. With this model he proposed a possible solution for a psychophysical phenomenon, known as repetition blindness, based on the timing of waves of excitatory and inhibitory spikes (Connolly & Reilly 2005).

He has also written a spiking neuron simulator, called Milligan, which facilitates exploring binding in CAs by means of synchronous spiking. The simulator can quickly process large numbers of spikes with millisecond accuracy (Connolly, Marian, & Reilly 2004).

Middlesex University School of Computing Science The School of Computing Science at Middlesex University was established in 1994. Middlesex has over 50 research active academic staff, 50 PhD students, and fourteen visiting research fellows including seven international visiting professors. The School also has staff dedicated to website design, development, and maintenance who will assist with the project website.

Huyck is head of the AI research group. A large number of faculty have expertise in neural nets (e.g. Dr Siri Bavan, Dr Ian Mitchell and Dr Usama Hasan), sensing (Dr Carl Evans, Dr Peter Passmore, Dr Xiaohong Gao and Dr Shaehedur Rahman), Visual Psycho-Physics (Dr Dan Diaper), Agent Technologies (Dr Satinder Gill), NLP (Dr Elenor Maclaren), and Robotics (Prof Anthony White). All can contribute expertise to the project.

References

Belavkin, R., and Ritter, F. 2004. Optimist: A new conflict resolution algorithm for act-r. In *Proceedings of the Sixth International Conference on Cognitive Modelling.*

Belavkin, R. 2001. Modelling the inverted-U effect in ACT–R. In Altmann, E.; Cleeremans, A.; Schunn, C.; and Gray, W., eds., *Proceedings of the 2001 Fourth International Conference on Cognitive Modelling.*

Belavkin, R. 2005. Entropy and information in models of learning behaviour. *AISB Quarterly* 119:5.

Connolly, C., and Reilly, R. 2005. A proposed model of repetition blindness. In *Modelling language, cognition and action*, volume 16. World Scientific. 279–88.

Connolly, C.; Marian, I.; and Reilly, R. 2004. Approaches to efficient simulation with spiking neural networks. In Bowman, H., and Labiouse, C., eds., *Connectionist models of cognition and perception II.* World Scientific. 231–240.

Gaizauskas, R.; Hepple, M.; and Huyck, C. 1998. Modifying existing annotated corpora for general comparative evaluation of parsing. In *Workshop on Parsing Evaluation: The First International Conference on Language Resources and Evaluation.*

Huyck, C. 1994. *Plink: An Intelligent Natural Language Parser.* Ph.D. Dissertation, The University of Michigan.

Huyck, C. 1998. The MUC7-plink system. In *Proc. of the Seventh Message Understanding Conference (MUC-7).* Morgan Kaufmann.

Huyck, C. 2000. A practical system for human-like parsing. In Horn, W., ed., *Proceedings of the 14th European Conference on Artificial Intelligence.*

Huyck, C. 2001a. Cell assemblies as an intermediate level model of cognition. In Wermter, S.; Austin, J.; and Willshaw, D., eds., *Emerging Neural Architectures based on Neuroscience.* 383–397.

Huyck, C. 2001b. Natural language parsing with cell assemblies: A model of nonconscious human language processing. In *Proc. of the AISB'01 Symposium on Nonconscious Intelligence: From Natural to Artificial*, 31–42.

Huyck, C. 2004. Overlapping cell assemblies from correlators. *Neurocomputing* 56:435–9.

Huyck, C. 2005. Variable binding of cell assemblies with binding areas and spontaneous neural activation. In *Proceedings of the 22nd Workshop of the European Society for the Study of Cognitive Systems.*

Huyck, C. submitted. Creating hierarchical categories using cell assemblies. *Connection Science.*

Huyck, C., and Bowles, R. 2004. Spontaneous neural firing in biological and artificial neural systems. *Journal of Cognitive Systems* 6:1:31–40.

Huyck, C., and Orengo, V. 2005. Information retrieval and categorisation using a cell assembly network. *Neural Computing and Applications.*

Kenny, I., and Huyck, C. 2005. An embodied conversational agent for interactive videogame environments. In *Proceedings of the AISB'05 Symposium on Conversational Informatics for Supporting Social Intelligence and Interaction*, 58–63.

LeThan, H.; Abeysinghe, G.; and Huyck, C. 2004. Generating discourse structures for written texts. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics.*

Lytinen, S.; Bhatacharyya, S.; Burridge, R.; Hastings, P.; Huyck, C.; Lipinsky, K.; McDaniel, E.; and Terell, K. 1992. The link system: MUC-4 test results and analysis. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 159–63. Morgan Kaufmann Publishers.

Lytinen, S.; Burridge, R.; Hastings, P.; and Huyck, C. 1993. Description of the link system used for MUC-5. In *Proceedings of MUC-5*, 293–304. Morgan Kaufmann Publishers.