Research Proposal:

Cognitive Approaches to Learning in a Hybrid Semantic Network: An Examination of Localist and Dual Localist–Distributed Representations and their Role in Concept Formation

Peter Dreisiger
Maritime Operations Division, Defence Science and Technology Organisation
Computer Science & Software Engineering, The University of Western Australia
prd@csse.uwa.edu.au

August 2008

A Proposed Study

A.1 Project Title

Cognitive Approaches to Learning in a Hybrid Semantic Network: An Examination of Localist and Dual Localist–Distributed Representations and their Role in Concept Formation

A.2 Background

Semantic networks are structures that represent knowledge using patterns of interconnected nodes and links; typically, the nodes correspond to entities and concepts, while the links define how they are related. In the field of artificial intelligence, these networks have been used to build expert systems, and for natural language processing; due to their intuitive nature, and their scale-free structure (Steyvers and Tenenbaum, 2005), they have also been used to implement efficient associative memory systems, and context-dependent forms of reasoning.

Within the cognitive sciences, semantic networks have been used to model the organisation of knowledge in memory (Collins and Quillian, 1969), to account for the structures and mechanisms of human working memory (Cowan, 1999), and in studies of phenomena including associative recall, priming and episode blending (Grinberg and Kokinov, 2003). They have even been used in the study of analogies and analogical reasoning (Holyoak and Thagard, 1989; Kokinov, 1994; Kokinov and Petrov, 2001).

The networks' versatility is due, at least in part, to their diversity. In his treatise on this subject, John Sowa (1992) identified five basic types of semantic networks: definitional, assertional, implicational, executable and learning; included in the last of these categories were the artificial neural networks (ANNs). While semantic and neural networks are both graphical forms of knowledge representation, there are some important differences between them; in particular, they differ in: (1) how they represent entities

and concepts, (2) how they represent properties and relationships between entities, and (3) the nature of the latent similarities.

In a conventional semantic network, knowledge is represented locally — that is, there is a one-to-one correspondence between concepts and nodes, and related concepts are tied together by typed links; associative, or weighted, links may also be used to provide apriori estimates of similarity. The *latent*, or implicit, similarity between two nodes is often defined in terms of the features that they share (see, for example, Tversky, 1977; Sun, 1995); some semantic networks, particularly those used in models of analogy, also consider nodes to be similar if they participate in the same kinds of relations (Gentner, 1983).

In contrast to these *localist* representations, concepts in an ANN are *distributed* over a large number of nodes, many of which are also used to represent other concepts; the concepts, and the relationships between them, are reflected by the weights of the links, and in the patterns of activation. An important property of these networks is that concepts which share correlated sets of features, or participate in similar sets of relations also tend to have similar representations; thus, in a neural network, the estimates of similarity may be latent, but they are always a function of the concepts' geometry, or location in the network's state space.

While neural networks can be used to develop distributed representations that reflect conceptual similarities, they are less well suited to learning large amounts of declarative knowledge for several reasons. Firstly, the learning process tends to generalise the relationships, and thus introduce encoding errors; also, there is no *direct* way to retrieve these relations once they have been learnt (rather, the retrieval process actually involves a series of queries that return the relations' bindings). And finally, it is particularly difficult to perform symbolic manipulation, or implement relational and pragmatic constraints, within an ANN.

In this project, we are going to examine how these two types of networks can be combined to form a hybrid, or dualistic, network that has the storage properties of a semantic network and the descriptive powers of a neural network.

A.3 Contribution to Scholarship

While the application and classification of semantic networks have been subjects of research since the 1960s (Sowa, 1992), the areas of network generation and growth have received comparatively less attention. This study will focus on the issues of automatic network generation, and concept formation in particular; specifically, it will develop, and characterise, a cognitively-feasible model of learning and concept formation within a hybrid semantic network.

In the first phase of the project, we will develop a dualistic semantic–neural network. This network will consist of two layers — a traditional semantic network that represents facts about entities using localist nodes, and a neural network layer that will generate distributed representations of each entity based upon these facts; similarities between the distributed representations can then be used to update the weights of associative links in the semantic layer, and to identify clusters that may correspond to new concepts or relations.

The use of neural networks, distributed representations and semantic *information* is not without precedent: they have been used to learn the structure of semantic relations (Dyer et al., 1992) and case–role representations of simple stories (Miikkulainen, 1993), and they have also been used in studies of attribute co-occurrence and conceptual similarity (Rogers and McClelland, 2004). This project is novel, however, in that it combines semantic and neural *networks*, and uses the resulting distributed representations to determine explicit measures of similarity; the dualistic network is also unique because it combines the

capacity and accuracy of a semantic network with the generative abilities of a neural network.

In the second phase of this project, we will examine the distributed representations to determine the nature of the concepts and relationships, and the types of similarity that can be captured by a dualistic network.

Within a semantic network, entities can be defined in terms of their properties, or micro-features, and the relationships that they participate in; these micro-features may, in turn, describe facts at different levels of abstraction — for example, perceivable attributes, functional capabilities, or taxonomic information. In this phase, we will also examine how sensitive the clusters are to changes in the *types* of micro-features that are included. The latent clusters will also be compared to those found using models of analogy, traditional associative algorithms and strictly localist representations.

To perform these analyses, we will develop a set of prototypical episodes, and a tool that can automatically generate specific episodes from these generalised scripts and entity descriptions; by defining classes of entities, and probability distributions over sets of features and events, this tool will allow us to rapidly generate a large number of training and test episodes. While a similar technique has been used to generate a training corpus from a set of sentence templates (Miikkulainen, 1993), the use of an automated episode generator in a study of working memory, analogy or concept formation has not been previously reported.

In the final phase of this project, we will use the results of our analyses to develop criteria for *what* defines a new concept, *when* they should be formed, and how changes in one layer should be propagated to the other. We will also assess the computational complexity of these criteria, and the effects that temporal delays and error tolerances have upon the concepts being formed and the average computational complexity.

B Research Plan

B.1 Background

In the field of artificial intelligence (AI), the term 'intelligent agent' usually refers to a software entity that can observe its environment and act in a goal directed manner; an agent is said to be 'rational' if it chooses actions that it believes will maximise its measure of performance, given its model of the world and its background knowledge. According to this definition, rational behaviour ought to be quite achievable; practically, however, there are still real differences between the way *we* act, and the way in which artificial agents behave.

Even in complex, real-world situations, we humans are able to infer a lot about our environment, and thus build up a detailed world model — a task that helps us to deal with uncertain, incomplete, and even erroneous information, but a task that tends to require context-specific knowledge. Exactly how we can give an artificial agent access to this sort of information is a subject of ongoing research, but previous studies have looked at modelling and representing context using logic, production rules, and local, or context-sensitive, knowledge bases (for an overview of these approaches, see Brézillon, 1999; Lorins et al., 2004; Turner, 1998).

Broadly speaking, an artificial agent may obtain this information in two ways: (1) by retrieving it from a static set of rules or situational descriptions, or (2) by generating a description of its current situation from a set of related cases or precedents. Given a suitable description, the first approach can provide an agent with a detailed and accurate description of its environment; however, this method usually requires a hand-crafted knowledge base. The alternative, while more prone to generalisation errors, is better able to handle new situations and 'noisy' information (see, for example, Turner, 1998). Even this approach,

however, defines knowledge at a reasonably coarse level — usually in terms of recurring, and recognisable, subsets of an agent's environment.

In this project, we are going to focus on knowledge representation and formation at the *conceptual* level. We will look at how cognitive models of human memory and analogy-making can be used to represent and retrieve context-dependent knowledge, and we will develop, and characterise, a cognitively-feasible model of concept formation within a model of human working memory.

B.1.1 Models of Human Working Memory

In cognitive psychology, the term 'working memory' refers to the structures and mechanisms that maintain task-relevant information in a highly accessible form for the duration of a cognitive task. In this sense, working memory is similar to its theoretical predecessor, short-term memory (STM).

Unlike STM, whose primary role is to *store* information, working memory is a more active and processing-oriented construct. It has been described as the 'workspace' or 'blackboard' of the mind, and it is where the active processing and temporary storage of task-relevant information take place (Miyake and Shah, 1999). Such a view of working memory requires a more sophisticated account of the control mechanisms that goes beyond simple memorisation strategies — mechanisms that explain psychological phenomena such as habituation, attention, priming and accessibility.

Models of working memory can be divided into two main categories — the multiple-component models, and the activation-based models. The former treat working memory as a plural construct that is separate from long-term memory (LTM), and have sense-specific stores and maintenance procedures (Baddeley and Logie, 1999). Their focus is on how we maintain *perceptual* information, and to a lesser extent, how we prepare for specific types of actions; they do not really account for how we maintain and manipulate declarative knowledge.

In contrast, the activation-based models treat working memory as an activated subset of LTM that is governed by a single, sense-independent, mechanism; they also assume that working memory is hierarchical, and that there is a strong correlation between activation and accessibility (Cowan, 1999). Most importantly for us, they are well suited to computer modelling. Indeed, these cognitive models underly, or are at least consistent with, those semantic networks that use spreading activation to identify context-specific information; they are also compatible with several models of analogy.

While the activation-based theories describe the structure of working memory and the mechanisms of selective attention, they do not explain *how* we identify similarities between concepts, or correspondences between situations; this *is*, however, the focus of cognitive models of analogy and analogical reasoning.

B.1.2 Analogical Reasoning

As with working memory, many models of analogical reasoning are built upon semantic networks. In fact, there has been a steady increase in the complexity of these networks; they have also become a more central part of the retrieval and mapping processes. The earliest computational models of analogy used definitional networks to represent simple situations and taxonomic information about the participants (Winston, 1980); mappings between situations were then found by brute force, using these networks and Tversky's (1977) measure of similarity.

The Analogical Constraint Mapping Engine (ACME) of Holyoak and Thagard (1989) used two semantic networks to satisfy the authors' structural, semantic and pragmatic constraints. The first was a constraint satisfaction network (CSN) whose nodes represented potential mappings between situations, and whose

links identified compatible sets of hypotheses; the second network contained the taxonomic and associative information, and was used to insert links into the CSN, between conceptually similar elements.

A more recent model of human working memory and analogy is the Associative Memory-based Reasoning (AMBR) system of Kokinov (1994) and Kokinov and Petrov (2001). Unlike the previous models, which simply *use* semantic networks, AMBR is implemented *within* a hybrid, declarative and executable network — that is, one where each node represents a piece of knowledge *and* performs a simple set of operations.

Where ACME uses a general knowledge semantic network to augment its representation of a situation, AMBR embeds its percepts and pragmatic constraints directly into its network. Where other models treat analogy as a sequential process that starts with retrieval, and is followed by mapping and transference, AMBR performs these stages concurrently; this means that the mappings can be used *as* they are brought into working memory.

In spite of their strengths and adaptability, semantic networks (and many of the applications that use them) still assume that the concepts, and the relationships between them, are relatively static and known ahead of time. Even when this assumption is true, the networks still need to be constructed — a task that is often carried out by hand, though there are, of course, a few exceptions to this rule.

In natural language processing, semantic networks are usually generated directly from a corpus; in these networks, the nodes represent words or their root forms, and the links between them reflect the frequency of their co-occurrence. Examples of network construction outside of natural language processing are much less common. One of the few models of working memory that is able to modify its own network is AMBR; while it is not able to acquire new *concepts*, it does use a simple measure of co-occurrence to create new associative links, and to update the weight of existing ones.

AMBR

While a detailed review of AMBR is beyond the scope of this proposal, we will briefly touch on three of its properties that are relevant to this project and its cognitive focus. Firstly, even though AMBR was developed to model the spontaneous use of analogy in problem solving, it is also one of the more complete models of activation-based working memory: intrinsically, it is hierarchical, it has a focus of attention that can be directed by internal and perceptual processes, and its retrieval process is affected by the reasoner's awareness, or internal context. Secondly, AMBR is one of the more psychologically *accurate* models of working memory; it has even been used to study the effects of context and priming on recall, problem-solving, re-representation and episode blending (Kokinov and Petrov, 2001).

Finally, it is unique in the way that it represents episodes — unlike other models, which tend to treat episodes as rigid and indivisible sequences of events, AMBR uses a loosely-knit, or decentralised, representation; not only is this more psychologically plausible, it actually gives AMBR the ability to retrieve past episodes and find analogies in an online, or continuous, environment. This last property, in particular, makes AMBR the ideal basis for a cognitively-inspired artificial memory system that represents knowledge at the conceptual level; what it lacks is the ability to acquire new concepts.

Starting with an implementation of AMBR, and using novel techniques and a neural network-based model of learning, this project will look at the issues of semantic clustering and concept formation within a semantic network.

B.2 Project Aim

To develop, and characterise, a cognitively-feasible model of learning and concept formation within a hybrid semantic network.

The specific aims are:

1. to develop a dualistic semantic–neural network that combines a hybrid semantic network and a multi-layer feed-forward neural network;
2. to develop a set of episode prototypes, and a stochastic episode-generation tool that will use them to produce additional sets of training data;
3. to characterise the distributed representations, and the relationships between them, *as* they develop, and for different types of micro-features;
4. to compare the *latent* clusters in the distributed representations to those found using models of analogy, traditional associative learning algorithms and strictly localist representations; and
5. to develop, and assess, criteria for the formation of new concepts.

B.3 Methodology

1. **Aim:**
   To develop a dualistic semantic–neural network that combines a hybrid semantic network and a multi-layer feed-forward neural network.

   Background:
   This project, and our use of a neural network to develop distributed concept representations, builds upon the three specific techniques: the *Distributed Symbol Discovery through Symbol Recirculation* of Dyer et al. (1992), Miikkulainen and Dyer's (1987; 1988) *Forming Global Representations with Extended backPropagation (FGREP)*, and Rogers and McClelland's (2004) *Backpropagation-to-Representation*.

   Dyer et al.'s system consists of a pair of multi-layer feed-forward neural networks (called short- and long-term memory), a distributed symbol store (DSM), and the set of relational expressions that describe a sample semantic network; the STM encodes the relational associations of each source node in the semantic network, while the LTM encodes the network as a whole. For example, given a set of expressions:

   $$((node_1\ ((relation_1\ node_3)(relation_2\ node_2)))\\ (node_2\ ((relation_2\ node_1)))\ \ (node_3\ ((relation_1\ node_2))))$$

   the STM network is first trained to learn the relational bindings for $node_1$ (i.e. $relation_1 \mapsto node_3$ and $relation_2 \mapsto node_2$). Standard backward error propagation is used to train the network; its inputs are set to the binary unit vector that corresponds to the relation, and the *current* pattern-of-activation corresponding to the destination node is retrieved from the DSM. After these associations have been learnt, the network's weights (in this case, W_1) are saved, and the procedure is repeated for $node_2$ and $node_3$.

   Once all of the sets of weights W_i have converged, they are used to train the auto-encoding LTM; after the LTM has learnt all of the STM weights, the patterns of activation across its hidden layer are then used to update the DSM. Finally, these new patterns are recirculated back to the STM, and the entire process is repeated until the distributed symbols converge. Essentially, the patterns of activation in LTM are a 'compressed' representation of their respective STM weights, and the circulation process forces similar concepts to develop similar representations. (See Figure 1).

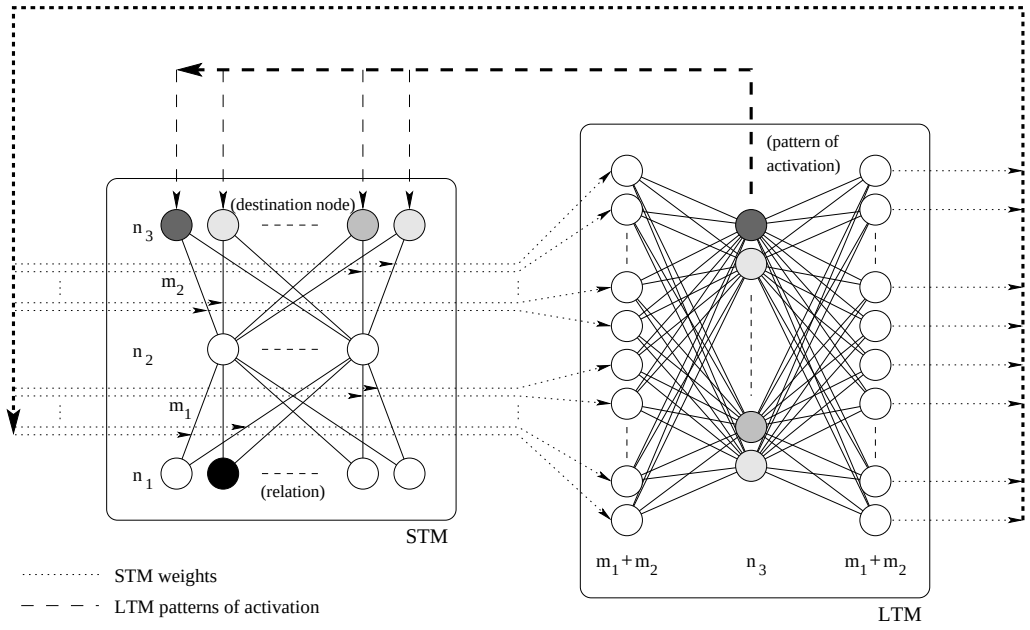

Figure 1: The basic structure of the symbol recirculation network, after Dyer et al. (1992). The STM network encodes the relation–binding pairs for each concept, while the LTM encodes the STM's weights across *all* concepts. The patterns of activation across the LTM's hidden layer develop into the concepts' distributed representation.

While this technique may be reminiscent of our approach, it is different in several respects: firstly, in their system, the semantic network serves only as a source of *training* data. Secondly, because the relational structures are stored in the neural network, the semantic information is susceptible to generalisation errors; also, the structure of the relationships can only be recovered by querying the neural network about every possible pair of concepts and relations.

Thirdly, in contrast to the entities' representations, the relations are represented by an orthogonal, binary set of activations — this means that the system can only learn about as many relations as there are nodes in the STM's input layer. Finally, and perhaps most significantly, it is unclear if this technique can learn to represent concepts whose relations have multiple bindings *without* introducing more errors[1].

Miikkulainen and Dyer (1987, 1988) took a somewhat different approach in developing FGREP. Like Dyer's earlier approach, FGREP uses a multi-layer feed-forward neural network. Unlike symbol recirculation, FGREP does not contain the equivalent of a STM; rather, changes to its distributed representation are achieved by propagating the error signals back, past the hidden layer, to the input layer. The FGREP architecture itself consists of a three-layer backpropagation network, and an external symbol store, or lexicon.

In Miikkulainen's (1993) extension to this approach, the network is trained to map syntactic representations of sentences to their corresponding case–role assignments. Its input and target patterns

[1] I.e. for semantic nodes of the form $(node_i\ ((relation_j\ node_k)(relation_j\ node_l)))$

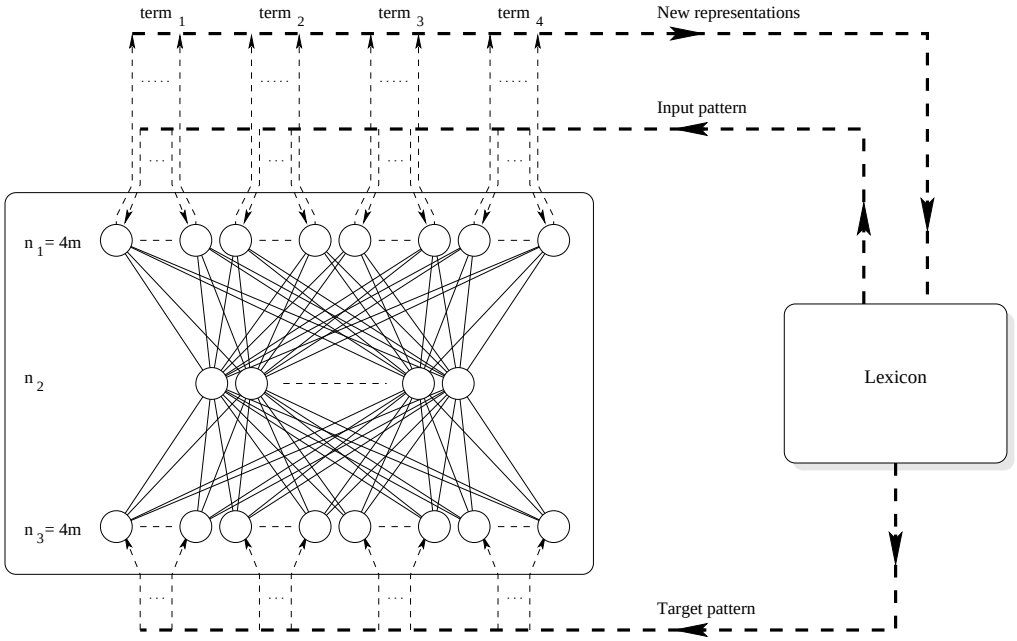

Figure 2: The basic FGREP architecture, after Miikkulainen (1993). This system consists of a three-layer feed-forward network, and an external lexicon to store the m-dimensional distributed representations. In this figure, the network is being taught to map a four-term (or symbol) input sequence to a four-term target sequence; at the same time, the representations used to construct the sequences are being updated by propagating the errors back to the input layer.

are generated from the syntactic and case–role assignments by retrieving the corresponding symbols from the lexicon, and concatenating their distributed representations (see Figure 2).

Training is accomplished by presenting the input pattern to the input layer, and the target to the output layer; the differences between the network's output and the target pattern are then propagated, through the hidden layer, back to the input layer. Finally, the new representations are loaded back into the lexicon, replacing the old ones. Because the input and target representations include a subject and an object (which are roughly equivalent to the originating and destination semantic nodes in Dyer et al.'s system), FGREP's multi-layer network is able to encode all of the mappings without the need for an intermediate, or STM, network.

While FGREP is able to capture many of the words' semantic features, it has trouble distinguishing between synonyms; even for a relatively small vocabulary, words with similar meanings tend to develop almost identical representations. To get around this problem, Miikkulainen added an unique, identifying pattern to each FGREP representation. In our project, however, this 'blurring' is not an issue as it is the semantic network (and not the neural) which differentiates between similar concepts and entities.

The last of the three techniques is called backpropagation-to-representation, and it borrows from the works of Miikkulainen (1993) and Dyer et al. (1992). Like FGREP, this technique develops its distributed representation by propagating the error signals back to the network's first distributed layer.

In terms of its inputs and outputs, however, backpropagation-to-representation is closer to Dyer et al.'s symbol recirculation technique than FGREP. While the network is trained using a series of item–relation–attribute tuples that resemble FGREP's subject–object assignments, the input and output layers in *this* approach consist of *localist* nodes; for this reason, Rogers and McClelland's system does not require a distributed symbol store.

Finally, like Dyer et al.'s technique, this approach is also more interactive than FGREP — that is, in order to retrieve information from the network, its inputs need to be bound to a specific item and relation pair; only then does the correct (or, in some cases, generalised) attribute appear at its output.

The approach that we are going to develop in this project lies somewhere between the works of these authors: like Dyer et al.'s symbol recirculation, our network will be trained to auto-associate information derived from a semantic network; like FGREP, however, our system will construct the input/target patterns by retrieving the distributed representations from an external store, and concatenating them.

And while our *network* is closer to these two systems than Rogers and McClelland's, our inclusion of perceptual representations, and the techniques that we will use to analyse the distributed representations, borrow more from their work than the two earlier studies.

Methodology:
The development of a dualistic semantic–neural network will consist of the following steps:

- *Implement AMBR 3 in C++.* While LISP sources are available for an earlier version of AMBR, we will be implementing AMBR 3 in C++ because changes will need to be made to couple AMBR to the neural network, and during Stage 4, when different localist forms of clustering will be investigated. In this step we will:
  - Implement DUAL (AMBR's underlying cognitive architecture), and its basic micro-agents, as per Petrov (1998) and Kokinov and Petrov (2001);
  - Implement an initial run-time environment with support for parsing and network construction;
  - Implement AMBR's micro-agents and structure-correspondence, constraint satisfaction, promotion and skolemisation mechanisms on top of DUAL; and
  - Validate the implementation according to the episodes and processing traces listed in Petrov (1998).
- *Implement FGREP in C++.* In this step we will:
  - Select an open-source ANN library based upon their ease of integration, their speed, and the availability of compatible visualisation and analysis tools.
  - Define a mapping from AMBR's representation of multi-place relations to the fixed-length format that will serve as the network's inputs and outputs; and
  - Implement an auto-encoding backward error-propagation network using the chosen ANN library.
- *Integrate the semantic and the neural network layers.* To do this, we will also need to define a programmatic interface between AMBR and the neural layer that will allow it to:
  - retrieve information about the concepts, and their relationships, from the semantic network;
  - store and retrieve its distributed representations from AMBR's conceptual micro-agents; and

– be notified of any changes to the *structure* of the semantic network.

2. **Aim:**
To develop a set of episode prototypes, and a stochastic episode-generation tool that will use them to produce additional sets of training data.

Background:
When it comes to producing the data sets needed to construct a semantic network or train a neural network, we have three options — we can: (1) compile them from existing corpora, (2) create the data sets by hand, and (3) use a utility to generate the data sets for us. Due to the sheer number of machine-readable documents, the first option is often used to train, and assess, natural language processing systems. The second option is more commonly used to study cognitive models of human working memory, reasoning and analogical recall; by hand-crafting a few well-known scenarios, a model's performance can be compared against other systems, or human subjects.

In the third option, a separate utility is used to create a large set of synthetic data. Typically, the data will be generated from a set of templates according to some sort of probability distribution; noise may also be introduced into the data set. This approach is often used to train and test neural networks, and it is the approach that we will be adopting in this study[2].

In this study, we are going to adopt the latter options for two reasons. Firstly, there are far fewer semantic and episodic knowledge bases and reference data sets than there are text corpora (instead, most studies of working memory, analogical reasoning and distributed representations tend to include their own data sets). And secondly, while systems such as AMBR can be downloaded together with their knowledge bases, the majority of their content is conceptual — they do not include the kinds, or amount of perceptual information that we will use.

For these reasons, we are going to create a set of prototypical episodes, and develop a utility to generate the larger training sets. Like Miikkulainen's sentence generator, our episode generation tool will accept, as its inputs, a set of scenarios, and a set of noun categories. Unlike the sentence generator, our utility will also support probabilistic variations in the scenarios, and the participating entities; specifically, it will support:

- the definition, and selection, of multiple *types* of micro-features (such as perceptual, functional, taxonomic and spatial);
- alternative and optional sequences of events;
- probabilistic definitions of the environment; and
- random variations in categories of the entities, and their perceivable properties.

The episode generator will also have access to the semantic and neural networks' clocks so it can simulate the passage of time.

Methodology:
The development of the episode-generation tool will consist of the following steps:

- Define an episodic and entity description language;
- Create a set of episode prototypes and entity definitions, including descriptions of the environment and entities' features;

[2]A simpler version of this technique was also used by Miikkulainen to generate the data needed to train his FGREP network, and to develop its distributed representations; using just 19 sentence templates and 12 noun categories, his utility was able to generate from 210 to over five million sample sentences (Miikkulainen, 1993, p. 76).

- Define a programmatic interface between AMBR and the generator that will allow it to (a) insert perceptual accounts of the episodes into the semantic network, and (b) retrieve feature and conceptual information;
- Implement a simple episode generator, with support for pre-defined and interactive scenarios; and
- Integrate it into AMBR's run-time environment.

3. **Aim:**
To characterise the distributed representations, and the relationships between them, *as* they develop, and for different types of micro-features.

Background:
Previous studies of distributed representations have used a combination of visual and statistical techniques to examine the symbols, and the similarities between them.

Miikkulainen (1993) used grey-scale images, and scatter plots of individual dimensions, to visually compare patterns of activation at the input and output layers; self-organising feature maps were also used to analyse the distribution of concepts in their 12-dimensional space. From the graphical representations, he found that related concepts shared similar patterns of activation, and that ambiguous terms tended to divide their representations between possible meanings. From the feature maps, he found that similar concepts formed hierarchical clusters in 2-dimensional space; as the maps used space-filling curves to represent high-dimensional structure, these clusters tended to be rather complicated and irregular.

Rogers and McClelland (2004), on the other hand, used box-plots to visually compare the patterns of activation, and a combination of hierarchical cluster analysis and multi-dimensional scaling to analyse the distribution and development of concepts in their 8-dimensional space. Hierarchical cluster plots, in particular, were used to visualise the relationships between concepts, and to track the development of the representations, and their latent hierarchies, over time; this technique was even able to capture changes in the *nature* of the hierarchies, and how they went from representing shared features to conceptual similarities.

Methodology:
In this stage of the study, we will conduct a series of experiments to determine the nature of the conceptual clusters, and how they vary (a) over time, and (b) with the *types* of features that are used to train the neural network. The first of these experiments will make use of perceptual, or low-level, micro-features, and those relations that are neither taxonomic nor purely associative; subsequent runs will include one or more of the following types of features:

- taxonomic relations;
- functional information that describes what the entities are capable of;
- environmental information that describes entities and features that are peripheral to the episodes;
- spatial cues that identify important juxtapositions; and
- associative, or co-occurrence, information.

The episode generator will also be used to introduce noise in the form of missing and unexpected features.

For each run, we will examine the distributed representations visually, and through the use of hierarchical cluster analysis and self-organising feature maps. The rate at which these clusters form will be tracked; the derived clusters will also be analysed to determine the following measures:

- the extent to which the latent cluster hierarchy differs from the one used to generate the episodes;
- the number of entities that are incorrectly classified (with respect to this reference taxonomy);
- the variance, or dispersion, of the discovered clusters (including and excluding entities that belong to multiple categories); and
- the variance of the discovered clusters, taking these outliers into account; and
- the variance of the 'true' clusters — i.e. those defined by the reference taxonomy.

The clusters will also be analysed *qualitatively*. In particular, we will look at the order in which general, basic-level and specific clusters are formed (Rosch et al., 1976); we will also examine outlying entities, and attempt to account for their relative isolation.

4. **Aim:**
To compare the latent clusters in the distributed representations to those found using models of analogy, traditional associative learning algorithms and strictly localist representations.

Background:
While Rogers and McClelland (2004) examined the structure and development of distributed symbols from localist information, there have not been any analyses of how these symbols, or their latent clusters, compare to those found through purely localist means. In this section, we will touch on two techniques that may be used to identify clusters within a traditional semantic network: (a) conceptual clustering, and (b) Hebbian learning.

Within the broader field of artificial intelligence, conceptual clustering is a form of unsupervised learning that generates a single classification tree using only information about the entities' attributes. Combining conceptual clustering with an activation-based model of working memory could improve this method in at least two ways. Firstly, the use of spreading activation and thresholds can significantly reduce the number of nodes, or concepts that need to be analysed; and secondly, providing the algorithm with multiple, context-dependent sets of data could induce *multiple* hierarchies.

The fact that conceptual clustering is limited to using attribute-value pairs, however, means that this technique is unable to use the sorts of relational information that is found in a semantic network like AMBR.

Although Hebbian learning was originally developed to account for long-term potentiation between biological neurons, it has also been used as a rule to update link weights within artificial neural networks. In essence, Hebbian learning states that the simultaneous activation of a set of neurons causes them to become strongly associated; conversely, neurons that are activated asynchronously develop inhibitory links between them (Haykin, 1999).

While Hebbian learning is often used to train auto-associative, or content-addressable memory, its use outside of neural networks is less common. An example of a semantic network that does use a form of Hebbian learning to identify co-occurrences is AMBR; its implementation, however, is somewhat restricted. Firstly, only nodes whose activation exceeds a specific threshold are considered; secondly, only those links between the *most* active node, and the rest of working memory, are strengthened; and lastly, AMBR only *increases* the link weights — it does not weaken them or create inhibitory ones.

The use of Hebbian learning in a localist network may also be problematic. When we develop a distributed representation, we want to identify, and capture, features that are highly correlated. In a neural network, this allows us to form conceptual clusters; in a semantic network, this is more

likely to identify sets of related *episodes*. Thus, while Hebbian learning may complement a model of analogical reasoning like AMBR, the types of associations that it captures will, most likely, lead to significantly different types of clusters.

Methodology:
In this stage of the study, we will use Hebbian learning and purely localist information to identify conceptual clusters within AMBR's semantic network; these clusters, and the types of associations they capture, will then be compared to those found in Stage 3.

To do this, we will conduct another series of experiments; for each of the episodes used in the previous stage, we will evaluate rules that vary according to the following criteria:

- whether they update links between *all* members of working memory, or just those that involve the most active node in the network;
- whether they increase *and* decrease, or just increase the link weights;
- whether the link suppression is context-sensitive; and
- the form of the function used to update the link weights.

After each run, we will identify the clusters of highly-associated nodes; these clusters will then be analysed qualitatively, and compared to those found in the distributed representations. The computational complexities of the distributed and localist approaches will also be compared.

5. **Aim:**
To develop, and assess, criteria for the formation of new concepts.

Background:
While conceptual clustering and categorisation are subjects of ongoing research in the data mining community, the area of concept formation has received relatively little attention; moreover, those techniques that refer to concept formation tend to focus on the generation of a concept hierarchy or a classification scheme (Han and Kamber, 2006).

Within the cognitive sciences, however, the subject of concept formation has been the focus of more research, and many theories have been developed to account for the formation of concepts, and natural classes, in early childhood (see, for example, Rosch et al., 1976). Several theories, in particular, have emphasised the importance of strongly correlated *sets* of features (Rogers and McClelland, 2004; Gärdenfors, 2000).

In the final stage of this study, we will use the results of our analyses to define a concept in terms of these distributed representations, and we will develop concept formation criteria that are consistent with this definition.

Methodology:
During the development of these criteria, we will consider:

- the proportion of the reference concepts (i.e. those used by the episode generator) that they can capture;
- their sensitivity to variations in the types of micro-features that were used to generate the distributed symbols;
- whether they are able to assign ambiguous entities to multiple concepts;
- whether they can be used recursively to form conceptual hierarchies; and
- their computational complexity.

We will also look at how these new concepts can be inserted into the semantic layer of the network.

B.4 Originality

To ensure that this project does not replicate any existing research, an extensive literature survey was conducted in the areas of semantic network generation and distributed concept representations; theories of human working memory and analogy were also reviewed to determine what, if any, cognitive models of concept formation have been developed.

During these surveys, we found several projects that looked, individually, at the development of distributed symbols from localist information, the mechanisms of associative recall in humans, and the statistical properties of concepts. What we did not find, however, were any studies that combined these areas into a coherent model of concept formation within a semantic network.

B.5 Project Timeline

The estimated timeline for the project is shown in Table 1.

C Scholars

James L. McClelland. Department of Psychology, Stanford University, Stanford, CA 94305, USA. `jlm@psych.stanford.edu`

Risto Miikkulainen. Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712-0233, USA. `risto@cs.utexas.edu`

Michael G. Dyer. Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90095-1596, USA. `dyer@cs.ucla.edu`

Boicho N. Kokinov. Department of Cognitive Science and Psychology, New Bulgarian University, Sofia 1635, Bulgaria. `bkokinov@nbu.bg`

Alexander A. Petrov. Department of Psychology, Ohio State University, Columbus, OH 43210, USA. `apetrov@alexpetrov.com`

D Bibliography

A. D. Baddeley and R. H. Logie. Working Memory: The Multiple-Component Model. In *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, chapter 2, pages 28–61. Cambridge University Press, 1999.

P. Brézillon. Context in Problem Solving: A Survey. *The Knowledge Engineering Review*, 14(1):1–34, 1999.

A. M. Collins and M. R. Quillian. Retrieval Time from Semantic Memory. *Journal of Verbal Learning and Verbal Behavior*, 8:240–248, 1969.

N. Cowan. An Embedded-Processes Model of Working Memory. In *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, chapter 3, pages 62–101. Cambridge University Press, 1999.

M. G. Dyer, M. Flowers, and Y.-J. A. Wang. Distributed Symbol Discovery through Symbol Recirculation: Toward Natural Language Processing in Distributed Connectionist Networks. In R. Reilly

Stage	Task	Weeks	Dates
—	Conduct literature survey	20	Feb 08 – Jul 08
	Write a review of cognitive models of memory and analogy, focusing on their relevance to artificial agents	4	Jul 08 – Aug 08
	Present work and progress report for confirmation of candidature	3	Feb 09
	Prepare annual progress report	2	Feb 10
1	**Implement a dualistic semantic–neural network**	**21**	Jul 08 – Dec 08
	Implement DUAL	6	Jul 08 – Sep 08
	Implement AMBR on top of DUAL	4	Sep 08
	Implement FGREP	8	Sep 08 – Dec 08
	Integrate the semantic and neural networks	1	Dec 08
	(Holidays)	(2)	Oct 08 – Nov 08
2	**Implement a stochastic episode-generation tool**	**14**	Dec 08 – Mar 09
	Create the initial set of episode prototypes	2	Dec 08
	Define the scripting language and generator requirements	2	Dec 08 – Jan 09
	Implement the generator and the additional episodes	8	Jan 09 – Mar 09
	(Christmas/New Year holidays)	(2)	Dec 08 – Jan 09
2	Write a paper about the network and episode generator	4	Mar 09 – Apr 09
3	**Characterise the distributed representations and the relationships between them**	**27**	Apr 09 – Oct 09
	Select and setup the analysis tools	8	Apr 09 – May 09
	Create a more extensive set of episodes	2	May 09 – Jun 09
	Experiment 1: Analyse the distributed representations	6	Jun 09 – Aug 09
	Write up Experiment 1	9	Aug 09 – Oct 09
	(School holidays)	(2)	Jul 09
4	**Compare the localist and distributed clusters**	**27**	Oct 09 – Apr 10
	Implement the Hebbian learning algorithms within AMBR	2	Oct 09
	Experiment 2A: Analyse the localist clusters	6	Oct 09 – Dec 09
	Compare the localist clusters to the distributed ones	2	Dec 09 – Jan 10
	Write up Experiment 2A	4	Jan 10 – Feb 10
	Experiment 2B: Investigate sub-graph mining in working memory	4	Feb 10 – Mar 10
	Write up Experiment 2B	4	Mar 10 – Apr 10
	(Holidays)	(1)	Oct 09
	(Christmas/New Year holidays)	(2)	Dec 09 – Jan 10
5	**Develop and assess criteria for the formation of new concepts**	**15**	Apr 10 – Jul 10
	Develop definitions of a concept	2	Apr 10
	Implement ways of finding these concepts	4	Apr 10 – May 10
	Experiment 3: Analyse the concepts and their formation	3	May 10 – Jun 10
	Write up Experiment 3	4	Jun 10 – Jul 10
	(School holidays)	(2)	Jul 10
—	**Write the thesis**	**16**	Jul 10 – Nov 10
	(Holidays)	(1)	Oct 10

Table 1: Project timeline as of August 2008.

and N. Sharkey, editors, *Connectionist Approaches to Natural Language Understanding*, pages 21–48. Lawrence Erlbaum, 1992.

P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.

D. Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7(2):155–170, 1983.

M. Grinberg and B. Kokinov. Analogy-Based Episode Blending in AMBR. In B. Kokinov and W. Hirst, editors, *Constructive Memory*, pages 157–167. NBU Press, 2003.

J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Elsevier, 2nd edition, 2006.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1999.

K. J. Holyoak and P. Thagard. Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, 13: 295 – 355, 1989.

B. N. Kokinov. A Hybrid Model of Reasoning by Analogy. In K. Holyoak and J. Barnden, editors, *Analogical Connections*, volume 2 of *Advances in Connectionist and Neural Computation Theory*, pages 247–320. Ablex Publishing, 1994.

B. N. Kokinov and A. A. Petrov. Integration of Memory and Reasoning in Analogy-Making: The AMBR Model. In K. J. Holyoak, D. Gentner, and B. N. Kokinov, editors, *The Analogical Mind: Perspectives from Cognitive Science*, chapter 2, pages 59–124. MIT Press, Cambridge, MA, 2001.

P. Lorins, P. Brézillon, and A. Gonzalez. Context-based Decision Making: Comparison of CxBR and CxGs Approaches. In *Proceedings of the IFIP Internation Conference on Decision Support Systems*, pages 115–124, 2004.

R. Miikkulainen. *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. MIT Press, 1993.

R. Miikkulainen and M. G. Dyer. Building Distributed Representations Without Microfeatures. Technical report, Department of Computer Science, UCLA, 1987.

R. Miikkulainen and M. G. Dyer. Forming Global Representations with Extended Backpropagation. In *IEEE International Conference on Neural Networks*, volume 1, pages 285–292, 1988.

A. Miyake and P. Shah, editors. *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*. Cambridge University Press, 1999.

A. A. Petrov. *A Dynamic Emergent Computational Model of Analogy-Making Based on Decentralized Representations*. PhD thesis, Central and Eastern Eurpean Center for Cognitive Science, New Bulgarian University, July 1998.

T. T. Rogers and J. L. McClelland. *Semantic Cognition: A Parallel Distributed Processing Approach*. MIT Press, 2004.

E. Rosch, C. B. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem. Basic Objects in Natural Categories. *Cognitive Psychology*, 8:382–439, 1976.

J. F. Sowa. Semantic Networks. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*. Wiley, New York, 2nd edition, 1992.

M. Steyvers and J. B. Tenenbaum. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science*, 29:41–78, 2005.

R. Sun. Robust Reasoning: Integrating Rule-based and Similarity-based Reasoning. *Artificial Intelligence*, 75:241–295, 1995.

R. M. Turner. Context-Mediated Behaviour for Intelligent Agents. *International Journal of Human-Computer Studies*, 48(3):307–330, 1998.

A. Tversky. Features of Similarity. *Psychological Review*, 84(4):327–352, July 1977.

P. H. Winston. Learning and Reasoning by Analogy. *Communications of the ACM*, 23(12):689–703, December 1980.

E Facilities

The School of Computer Science and Software Engineering has committed to providing an office space with network and telephone facilities. Facilities, and additional computing hardware, are also available at the Defence Science and Technology Organisation site on HMAS Stirling.

E.1 Supervision

This project will be supervised by Dr Cara MacNish and Dr Wei Liu of the School of Computer Science and Software Engineering.

E.2 Special Equipment

No special equipment is required for this project.

E.3 Special Techniques

No special techniques are required for this project.

E.4 Special Literature

No special literature is required for this project.

E.5 Statistical Advice

Statistical advice is available from the Statistics Clinic run by the School of Mathematics and Statistics.

F Estimated Costs

The School of Computer Science and Software Engineering will provide an annual budget of $1,000.

G Fieldwork

This project does not involve any fieldwork.

H Supervisors

Coordinating supervisor:	Dr Cara MacNish. Senior Lecturer, School of Computer Science and Software Engineering, The University of Western Australia.
Areas of academic expertise:	Machine learning, neural networks, statistical analysis and biologically inspired computing.
Contribution to supervision:	50%

Co-supervisor:	Dr Wei Liu. Lecturer, School of Computer Science and Software Engineering, The University of Western Australia.
Areas of academic expertise:	Machine learning, ontology learning, natural language processing and autonomous agent design.
Contribution to supervision:	50%

I Confidentiality & Intellectual Property

This project is being funded by the Defence Science and Technology Organisation. As such, it governed by the Intellectual Property (IP) Agreement made between the Commonwealth of Australia and the University of Western Australia. The agreement itself was submitted with my application for postgraduate study, and a copy is on file with the Graduate Research School.

To summarise the terms of this agreement:

1. All developed IP shall be owned by the Commonwealth;
2. The University shall treat all developed IP as confidential and, subject to certain caveats, shall not disclose that information to any third party without the prior written consent of the Commonwealth;
3. The University may use the developed IP for research purposes only; any such use must not prejudice the Commonwealth's rights of use of the developed IP. Also, work based upon this developed IP shall be licensed to the Commonwealth for its own use without charge or restriction; and
4. The University shall not, without the prior written approval of the Commonwealth, publish or present any material which might prejudice the Commonwealth's national security or IP interests. If the University publishes material relating to the developed IP, an acknowledgement of the Commonwealth's role in providing background materials must be included in the publication.