Declarative Learning: Cognition without Primitives

Edmund Furse and Roderick I. Nicolson

Department of Computer Studies
The Polytechnic of Wales
Pontypridd
Mid Glamorgan
CD37 1DL
UK
efurse@uk.ac.pow.genvax

Department of Psychology
The University of Sheffield
Sheffield
S10 2TN
UK
r.nicolson@uk.ac.sheffield.primea

Abstract

Declarative learning by experience is a foundation cognitive capability, and we argue that, over and above the normal processes of declarative learning, the ability for truly novel learning is the critical capability which bootstraps human cognition. Next we assert that none of the established models of machine learning and no established architecture for cognition have adequate declarative learning capabilities, in that all depend for their success on some pre-characterisation of the learning domain in terms of state space or pre-existing primitives geared to the domain. Finally we describe briefly the Contextual Memory System, which was designed explicitly to support all five declarative learning capabilities. The CMS underlies the Maths Understander machine learning system which 'reads' mathematics texts from scratch, assimilating mathematics concepts, and using them not only to check proofs but also to solve problems.

Introduction

The origins of human knowledge have provided a fertile source of speculation over the millenia, and even today the issue is far from resolved. Meno's paradox, the 'learning paradox' derives from the ancient Greek sophists who argued that truly novel learning was impossible in that *"novel knowledge cannot be derived completely from old knowledge, or it would not be new. Yet the transcending part of it cannot be completely new either, for then it could never be understood."* (Boom, 1991, p274). Plato sidestepped this paradox by asserting that all knowledge was innate, but initially dormant, and the ensuing debate between empiricist and nativist positions has echoed down the centuries. Piaget (1952, 1985) tackled the issue by arguing that the process of equilibration, the striving to change cognitive structures to avoid cognitive disequilibrium, was the key to truly novel learning, and went on to argue that his four stages in cognitive development derived from three such qualitative changes in cognition. Piaget's stage theory has, of course, been frequently criticised, but his primary emphasis was on genetic epistemology, the origins of knowledge. Fodor (1980) has revived the Meno paradox, and adopted a nativist position, arguing that Piaget's equilibration concept was not powerful enough to support the acquisition of novel cognitive structures, and the issue remains unresolved in the developmental literature (see Boom, 1991 and Juckes, 1991 for recent analyses). Unfortunately, the debate has focused on development rather than learning, even though it is clearly nonsense to suggest that true learning occurs only at three or four stages in one's lifetime, and in reality children (frequently) and adults (sometimes) are confronted by the need to acquire knowledge in a completely new domain. Adults are able to acquire the concepts of new games, new academic subjects (mathematics, computer programming, statistics, geology, etc.), new social conventions. Children have to start from scratch. They are refuting the learning paradox most of the time in the early years. Consequently, we argue that the learning paradox is at least as great a challenge for cognitive science as it is for developmental psychology.

We use the term declarative learning to refer to all aspects of the learning of declarative knowledge. It has many facets, and in Furse and Nicolson (1991) we stated the necessary competences of a declarative learner, which we termed comprehension, assimilation, utilisation, and accommodation. We now make explicit the fifth requirement, that of supporting truly novel learning.

(i) Comprehension (cf. Encoding). The system must be able to encode novel information so as to cause it to enter working memory. We use the term comprehension to emphasise the need to recode the input into the format used in the declarative memory structures.

(ii) Assimilation (cf. Storage). The system must be able to create a long-term memory entry for information in its working memory. Furthermore, the system must be able to incorporate the new information into its memory structures in such a way as to facilitate the adaptive use of that information in other contexts. It is worth noting the need to store not only the new information but also the context in which it occurs. Tulving (1983) refers to this as the 'cognitive environment'.

(iii) Utilisation (cf. Retrieval). The system must be able to retrieve the stored information given an appropriate cue or an appropriate context. For many theorists, the term retrieval suggests too automatic a process, and terms such as reconstruction (Bartlett, 1932) or ecphory (Tulving, 1983) capture better the complex search and matching processes involved. We adopt the neutral term to indicate the adaptive use of the existing knowledge to satisfy the system's requirements.

(iv) Accommodation. Regardless of how broadly one interprets the above three processes, we believe that they are incomplete for a viable declarative learner. In addition the system must be able to modify the information in the light of subsequent experience so as to improve its adaptivity of use. This includes the making of new links, the strengthening of salient associations, the forgetting of useless associations, and the creation of new features.

(v) Novel Learning. While the above processes provide a reasonable description of much declarative learning, it is possible to envisage a declarative learner which showed all four competences but was unable to undertake truly novel learning. Consequently, as a fifth, stringent criterion for declarative learning, we argue that it is necessary to demonstrate learning 'from scratch' without any pre-characterisation of the space to be learned, or any pre-characterisation of the primitives required for learning in that domain.

Existing Cognitive Science Approaches to Declarative Learning

The problems in acquiring knowledge have recently moved centre stage in cognitive science. Anderson (1990) argued that the origins of human knowledge were one of the major issues for cognitive science, concluding by reduction that from the viewpoint of ACT*/PUPS the weak problem solving principles (by which all subsequent domain-dependent problem solving productions are derived) must themselves be innate. Lenat and Feigenbaum (1991) present as one of the guiding principles of AI the 'Knowledge Principle' — *"If a program is to perform a complex task well, it must know a great deal about the world in which it operates. In the absence of knowledge, all you have left is search and reasoning, and that isn't enough."* Indeed, Lenat argues that this principle mandates a new direction for the main enterprise of AI, and accordingly has devoted the last seven years to his ambitious CYC project for hand-coding a significant subset of human knowledge, thus providing what he hopes will be a knowledge-base sufficiently rich to bootstrap natural language understanding systems, and, ultimately, machine learning systems.

Encouragingly, the three major architectures for cognition — Soar (Laird et al, 1986), ACT* (Anderson, 1983) and connectionist approaches — have a learning mechanism as the cornerstone of their approach to cognition. ACT*/PUPS suggests that declarative knowledge must be acquired initially, then this declarative knowledge is 'proceduralised' by a 'knowledge compilation' process consequent upon successful performance, turning it into a production rule format. The production rules may subsequently be tuned by extended practice. Soar learns primarily by a process of problem-space search, with learning taking place by automatic processes of 'subgoaling' following a failure and by 'chunking'. Connectionist models learn by processes of differential link strengthening procedures based on learning procedures such as gradient descent (eg. Hinton, 1989).

Very surprisingly, summarising an analysis presented in Nicolson and Furse (1992), not one of the above cognitive architectures is able to cope with declarative learning.

Neither of the above symbolic architectures are able to cope with declarative learning, in that they both precharacterise the space in which learning is to take place. Anderson hand-crafts the appropriate declarative knowledge in each of the domains in which he works, and Soar makes the assumption that all learning can be characterised as search through a problem space, an assumption criticised by Boden (1988) as quite untenable.

The ability of connectionist models to perform at all at the symbolic level remains controversial. In brief, we argue that none of these approaches addresses the Learning Paradox.

One might expect that the problem of declarative learning would have been extensively studied in the machine learning literature, but, bafflingly, this is not so. Space precludes a detailed analysis of the machine learning literature here (see Ellman, 1989 for a useful review), and we shall merely note four problems of the established machine learning demonstrations. Thus four major critiques of the machine learning research are that most of the models lack psychological plausibility; that the declarative knowledge being acquired is relatively unstructured and semantically arid; that the learning tasks are too simple to be ecologically valid; and that all the models operate within a closed world in which the knowledge representations are pre-specified.

In summary, declarative learning is of major theoretical and applied importance, yet no established cognitive architecture offers any mechanism for it, and no current machine learning approach offers a principled and psychologically plausible account of the processes involved. Consequently, approaches to modelling human cognition are critically incomplete.

Overcoming the learning paradox- the Contextual Memory System

Let us start by trying to derive an informal requirements analysis for overcoming the learning paradox. Consider a person confronted by some novel event or situation — watching a new game; examining some complex, unfamiliar machine; or trying to understand some unfamiliar branch of mathematics or computer programming language. First impressions are of a mass of detail — pieces and actions; pipes, cables and bolts; or series of symbols. It is not initially a problem of telling the wood from the trees, it is a problem of even telling what the trees are! One can initially identify neither which features are salient nor the overall purpose of the components of the game, machine or language. An expert may help by labelling a few key components — the pieces, the components, or the commands. One stores this information, blindly, without understanding, but it forms the basis around which further information can be accreted, slowly building a better ability to describe the components and the key features. Eventually, over a period of time, one acquires the ability to tell the wood from the trees, understanding not only what the purpose of the components are, but also which features are salient.
Several approaches, including ACT*, Soar and connectionist approaches, give a reasonably plausible account of the later learning events, the accommodative processes which tune the existing knowledge to achieve better task performance, but the learning paradox arises in the initial stage — how can we take in new information when we understand neither what is relevant nor what features to look for?

We argue that one plausible approach is to generate as many features as possible for each event, and to attempt to maintain and adapt this population on the basis of subsequent events. Over time, the feature population will 'evolve' into one which fits better into the current evolutionary niche, in that salient features should emerge, and useless ones should die out.

This analysis suggests that one needs

(i) a mechanism for generating features automatically from input without the need for some prior characterisation in terms of domain-specific primitives

(ii) a mechanism for tuning the population of features on the basis of subsequent experience so as to encourage it to adapt to the domain.

This is made more formal in the Contextual Memory System which at the top level can be thought of as providing mechanisms for the storage and retrieval of information, both storage and retrieval being in terms of dynamic features. Information from the external environment is stored within the CMS as items which are indexed by features. The CMS starts with no items and no features. When an item is remembered, feature analysis takes place dynamically to create new features. This process of remembering will use a mixture of old features and new features. The features and items have energy which decays with time, but increases on recall. The links between the features and items have strengths which are adjusted on recall to ensure that the most salient features have the higher strengths. Since features are created dynamically the resulting memory configuration is complex and depends upon the history of memory processes.

When an item is first stored, it will be in terms of the currently high energy features. When subsequently recalled, it may be found with a different set of features, and a process of adjusting the CMS takes place to improve future access of the item. We now turn to how it is possible to generate the features dynamically without them being built in, and how the features are updated through experience.

Building features from the environment

It is possible to build features of new information without using built in primitives, by using built in feature building *mechanisms*, rather than built in features themselves. The essential idea is to break down the input into parts. The parts may not be meaningful to the agent on initial encoding, but through subsequent experience, these parts acquire meaning. We have found in the area of mathematics that the use of positional information is useful for subsequent retrieval of the information, and this also provides a simple model of attention. This process is best illustrated by means of an example from Winston's definition of a cup (Winston et al., 1983).
This is normally represented as:
cup(x)
⇔ liftable (x) and stable (x) and open-vessel (x)
In the FEL language (Furse, 1990), it would be represented as:
Definition of cup
x isa cup
iff x isa liftable and x isa stable and x isa open-vessel

In whatever manner "cup" is initially represented, it is not necessary to have already represented the notions of "liftable", "stable" and "open-vessel", or even "and", "⇔" in order to build features of the

object. This is most easily demonstrated by the mechanism used by the MU system (Furse, 1992a) and the current implementation of the CMS (Furse and Nicolson, 1991, Furse 1991, Furse 1992b), whereby the input is first parsed into a predicate calculus like representation:

```
(<=> (cup x)
     (and (liftable x)
          (and (stable x) (open-vessel x))))
```

We do not claim that the predicate calculus is used by people as an internal representation, this is just used for illustrative purposes. This datastructure can be thought of as a tree, and it is possible to analyse it into a number of features using various feature building mechanisms. The space of feature names is formally defined in BNF by:

```
<feature-name> ::= <pos><spec><type><term>
<pos>     ::= LHS- | RHS- | null
<pos>     ::= LHS-<pos> | RHS-<pos>
<spec>    ::= IS- | HAS-
<type>    ::= FORM- | TERM-
```

Most of this apparatus is to give a formal notion of the focus of attention, namely what part of the information the agent is attending to when generating the feature. The end of the feature-name, namely the <term> ensures that the feature space is infinite and only determined by its inputs, ie no pre-characterisation. LHS- means one is focusing on the left hand side of the tree, RHS- means the right hand side, and composition takes one further down left or right branches of the tree. HAS- means that the term occurs somewhere within the focus of attention, whilst IS- means that the term occurs exactly at the specificed focus of attention. A form is a canonical representation of a term useful in mathematics whereby the leaves of the tree are replaced by the letters a, b, c, ... to result in a canonical representation. Abstract HAS-forms introduce abstraction whereby parts of the tree are replaced by nodes. For a given term the number of abstract HAS-forms is in general very large.

Some of the mechanisms to generate features are:

Break the tree into single level HAS- forms without any positional information, eg

HAS-FORM-[LIFTABLE_A]
HAS-FORM-[STABLE_A]

Break the tree into single level HAS- forms with positional information, eg

RHS-HAS-FORM-[LIFTABLE_A]
LHS-RHS-IS-FORM-[STABLE_A]

Break the tree into abstract HAS- forms, at whatever level, eg to capture the notion that one has noticed that it is a definition of a cup:

HAS-FORM-[<=>_[CUP_A]_B]

One has noticed that there are three anded expressions on the right:

RHS-HAS-FORM-[AND_A_[AND_B_C]]

One has noticed the liftable and stable notions somewhere:

HAS-FORM-
[AND_[LIFTABLE_A]_[AND_[STABLE_B]_C]]]

Using these mechanisms it is possible to generate scores or often hundreds of different features of the input. This process is described in greater detail in Furse (1992b). Clearly the current implementation needs the use of a tree data structure, but mechanisms could be built to work on a string representation, for example one could break up

"x isa cup iff x isa liftable and x isa stable
and x isa open-vessel"

into features like:

HAS-FORM-"liftable"
START-HAS-FORM-"a isa cup"
HAS-FORM-"a isa liftable and a isa stable and b"
HAS-FORM-BEFORE-"liftable"_"open-vessel"
HAS-FORM-ADJACENT-
"a is liftable"_"a isa stable"

although one might want to allow the form abstraction to range over all substrings and not just the "x" as in the examples above, or not use it at all.

Learning the appropriate features

The processes described above allow the agent to generate a large number of features of the input information. But many of these features will be redundant. It is only through experience that the agent discovers which of the features are relevant for making future retrievals from memory. But it is essential that redundant features are generated at the outset, otherwise we are in danger of being in a closed box. Thus it is in the process of accommodation that the CMS models this learning experience.

In the CMS all features are given an energy value, and the features of highest energy can be thought of as representing the current attentional state. The CMS starts as a tabula rasa with no features, but as it stores information it generates features dynamically from the input as described above. The very first item will of course be represented in terms of brand new features, each of which will be given a default energy value. When subsequent items are stored, feature analysis will generate a mixture of old and new features. In the CMS a mixture of old and new features is stored, with the old features being the ones of highest energy. At the time of initial storage this may not be a very good choice of features for retrieval purposes, but provided that enough were generated and the agent is given useful subsequent experience, the features may subsequently be refined.

Learning in the CMS takes place during retrieval of information from memory. Any given probe gives rise to a number of features, and of course only the old features are used to index the memory. This search is implemented sequentially, but conceptually could be considered a parallel process whereby features activate items in memory that they index. Some

items will be indexed by more than one feature, and once their activation energy (technically within the CMS this is known in the retrieval process as a transient energy) rises above a threshold, the item fires and is tested against the probe. When the item is not in the memory this is discovered very fast. If it is present, retrieval will result in the found item which matches the probe, and a number of failures. Learning is then the processes of accommodation whereby the CMS memory structures are adjusted so that in future it is easier to retrieve the found item than the failures.

There are four accommodative processes: storing uncomputed features, increasing the energy of useful features, decreasing the energy of unuseful features, and creating new distinguishing features.

Uncomputed Features. Uncomputed features arise because not all the features that are found in the probe may be stored as indexing the recalled item. The context of recall may be very different from when the item was first stored and other contexts when it was recalled, so that at the time of recall it may not be known whether the item has one of the probe features or not. Features thus have a 3-valued logic: positive, negative or uncomputed; only positive feature links are stored in the CMS. Thus the features which have not been previously computed for the item, and are found to be positive for the item are now given new links to the item.

Useful and Unuseful Features. Features are deemed to be useful in the search if they index the found item, but not the failed items. These features can have their energies increased and also the energy of the link from the feature to the item. Conversely, features are considered unuseful if they index the failed items but not the desired item, and they have their energies decreased.

Creating new distinguishing Features. The final process of learning from the experience of recall is to dynamically create new features to distinguish the found item from the failures. In the CMS this is currently only done when no existing features succeed in doing this distinction. New features can be created by two different approaches. In the first the found item and a failure are analysed to discover differences, from which a feature can be derived. In the second approach both the found item and a failure are broken up into a large number of features, and a feature in the found item set and not the failure is chosen. The CMS currently uses the latter approach.

Through these processes of accommodation, the CMS continually adjusts itself with experience so that the items in memory that are the most important are most easily retrieved because specialised features will have been built and they will have high energy. Conversely items rarely needed take more time to retrieve because they may only have features of low energy, and furthermore several of these features may not be useful in any context other than the original presentation, ie they are redundant. This could be called the "Redundant Feature Hypothesis", and further empirical work is planned to test whether people need to store redundant features.

MU: the Mathematics Understander

While the CMS is intended as a generic architecture for learning by experience, it has to date only been thoroughly investigated in the domain of pure mathematics (Furse, 1992a). The CMS forms the basis of MU, a large computer program which models the reading of mathematics texts by students. MU has competence in all aspects of declarative learning, and is unique both in machine learning and in cognitive architectures in its ability to truly capture declarative learning.

To date, MU has been applied to two branches of pure mathematics, namely Group Theory (using the textbook by Herstein, 1975) and Classical Analysis (using the textbook by Anderson, 1969). The approach adopted involves first rewriting the text by

Definition 2.1.7
G is abelian iff G isa group and $\forall$ a,b a $\in$ G and b $\in$ G ab=ba

....

Problem 2.3.3
Prove (G isa group and $\forall$a,b $\in$ G $(ab)^2 = a^2b^2$)
$\Rightarrow$ G is abelian

Solution:
Suppose G isa group
and $\forall$a,b a $\in$ G and b $\in$ G $\Rightarrow$ $(ab)^2 = a^2b^2$
RTP G isa abelian
RTP G isa group and $\forall$a,b a $\in$ G and b $\in$ G
$\Rightarrow$ ab = ba by definition of abelian
Part 1
RTP G isa group
Follows logically
QED Part 1
Part 2
RTP $\forall$a,b a $\in$ G and b $\in$ G $\Rightarrow$ ab = ba
Suppose a $\in$ G and b $\in$ G
RTP ab = ba
Now $(ab)^2 = a^2b^2$
$\Rightarrow$ (ab)(ab) = (aa)(bb) since $x^2 = xx$
$\Rightarrow$ a((ba)b) = a((ab)b) since (ab)(cd) = a((bc)d)
$\Rightarrow$ (ba)b = (ab)b since au = aw $\Rightarrow$ u = w
$\Rightarrow$ ba = ab since ua = wa $\Rightarrow$ u = w
QED Part 2

Figure 1. Problem Solving in Group Theory by MU

hand into FEL — a 'formal expression language' (Furse 1990), which captures the essential semantics of the domain — thus forming a machine understandable text for MU to 'read'. The basic requirement in reading the text is to 'comprehend' each input line, and then for each section either to assimilate new knowledge, to check through each step of a proof, or to attempt to solve a problem. An example of successful problem solving is shown in Figure 1 (Definition 2.1.7 for abelian has been read in and assimilated earlier, as has the definition for group).

The key problem in understanding proofs and solving problems is to be able to find the appropriate mathematical result. MU uses the CMS to ensure that it does not suffer from a combinatorial explosion, by focusing its search to only inference rules that are relevant to the current context. Fig.1 shows an example of MU's ability to solve problems in group theory. When MU is trying to reason forwards from the step $(ab)^2 = a^2b^2$, it does a feature search using features such as:

IS-FORM-[=>_A_B]
LHS-IS-OP-=
HAS-FUNCTION-*
HAS-FUNCTION-=
RHS-LHS-IS-OP-*
HAS-FUNCTION-SQUARE

These features have partly already been refined through the experience of checking proofs. The features retrieve a number of relevant inference rules for example $x^2 = xx$, and also other inference rules which do not match the input. The CMS is then adjusted to ensue that the right inference rule can be found more easily in future.

Because of MU's extremely rich knowledge, all of which is learned, MU demonstrates much more complex mathematical understanding than nearly all programs derived from the theorem-proving tradition of AI, a paradigm case of Lenat's Knowledge Principle.

Conclusions

In conclusion, one of the major computational problems facing the human infant, and also the human adult, is how to make sense of new situations. This is surely achieved by dynamic, adaptive learning. We argue, therefore, that a dynamic, declarative learning capability should be the cornerstone of architectures for cognition. The CMS has provided an existence proof that a declarative learner can be constructed for the mathematics domain. We hope that this demonstration will prove the catalyst for the construction of a new generation of cognitive architectures.

References

Anderson, J. 1969. *Real Analysis.* Logos Press.

Anderson, J.R. 1989. A theory of the origins of human knowledge. *Artificial Intelligence*, 40: 313-351.

Boden, M. 1988. *Computer models of mind.* Cambridge: Cambridge University Press.

Boom, J. 1991. Collective development and the learning paradox. *Human Development*, 34: 273-287.

Ellman, T. 1989. Explanation-based learning: a survey of programs and perspectives. *ACM. Comp. Survey*, 21: 163-221.

Furse, E. and Nicolson R.I. 1991. The Contextual Memory System and Learning Mathematics, *AISB Quarterly Autumn/Winter 1991 no.78* (Special Issue on Hybrid Models).

Furse, E., 1990. A Formal Expression Language for Pure Mathematics, Technical Report CS-90-2, Department of Computer Studies, The Polytechnic of Wales.

Furse, E., 1991. Mathematical Expertise and the Contextual Memory System, Technical Report CS-91-6, Department of Computer Studies, The Polytechnic of Wales.

Furse, E., 1992a. The Mathematics Understander. *Proceedings of the International Conference on Artificial Intelligence in Mathematics*, Oxford University Press (in press).

Furse, E., 1992b) The Contextual Memory System: a cognitive architecture for learning without prior knowledge. *Cognitive Systems* (in press).

Herstein, I.N. 1975. *Topics in Algebra.* London, Wiley.

Juckes, T.J. 1991. Equilibration and the learning paradox. *Human Development*, 34: 261-272.

Lenat, D.B. and Feigenbaum, E..A. 1991. On the thresholds of knowledge. *Artificial Intelligence*, 47: 185-250.

Nicolson, R.I. and Furse, E. 1992. Declarative Learning: a Challenge for Cognitive Science. *Submitted.*

Piaget, J. 1952. *The origins of intelligence in children.* NY: International Universities Press.

Piaget, J. 1985. *The equilibration of cognitive structures.* Chicago: University of Chicago Press.

Winston, P.H., Binford, T.O., Katz, B., and Lowry, M. 1983. Learning physical descriptions from functional definitions, examples and precedents. In Proceedings of the American Association of Artificial Intelligence, (AAAI83), 433-439, Morgan Kaufmann.