Kirsh D. PDP learnability and innate knowledge of language. In: S Davis (Editor),
Connectionism: Theory and practice, The Vancouver Studies in Cognitive Science, Vol III
New York, Oxford University Press. pp 297-322, 1992

PDP Learnability and Innate Knowledge of Language

David Kirsh

Department of Cognitive Science, UCSD

ABSTRACT

It is sometimes argued that if PDP networks can be trained to make correct judgements of grammaticality we have an existence proof that there is enough information in the stimulus to permit learning grammar by inductive means alone. This seems inconsistent superficially with Gold's theorem and at a deeper level with the fact that networks are designed on the basis of assumptions about the domain of the function to be learned. To clarify the issue I consider what we should learn from Gold's theorem, then go on to inquire into what it means to say that knowledge is domain specific. I first try sharpening the intuitive notion of domain specific knowledge by reviewing the alleged difference between processing limitatons due to shartage of resources vs shortages of knowledge. After rejecting different formulations of this idea, I suggest that a model is language specific if it transparently refer to entities and facts about language as opposed to entities and facts of more general mathematical domains. This is a useful but not necessary condition. I then suggest that a theory is domain specific if it belongs to a model family which is attuned in a law-like way to domain regularities. This leads to a comparison of PDP and parameter setting models of language learning. I conclude with a novel version of the poverty of stimulus argument.

INTRODUCTION

It is widely assumed that PDP learnability has some bearing on questions of innateness. If a PDP network could be trained to make correct judgements of grammaticality, for instance, it seems to follow that innate knowledge of grammar is not necessary for language acquisition. The reason, quite simply, is that the learning rules used in PDP learning -- whether backpropagation or related gradient descent methods-- are general, domain independent methods. They are what AI theorists call weak methods. Hence in teaching a system to make correct judgements we seem to have an existence proof that there is enough information in the stimulus to permit learning by inductive means alone. It is this idea, and the methodological implications that flow from believing it, that I wish to explore here.

The problem I have with this argument is that to discover a network that will learn successfully designers must choose with care the network's architecture, the initial values the weights are set to, the learning rule, and the number of times the data set is to be presented to the network -- this latter parameter effects the smoothness of the estimated function. If such parameters are not controlled for, successful learning is extremely improbable. In thoughtful modelling, these parameters are chosen on the basis of assumptions about the nature of the function the system is to learn. That is, on the basis of assumptions about the task and the task domain. Prima facie, then, although the learning mechanism operating on data is a general one, the success of this mechanism depends equally on a set of antecedent choices that seem to be domain specific.

If these assumptions are genuinely domain specific we ought to reject PDP learnability as proof of inductive learnability. Learning can be viewed as a controlled process of moving from an initial state of knowledge about a domain to a more advanced state. The hallmark of true inductive learnability is that the initial state contains zero knowledge of the domain: all domain knowledge is acquired through learning. To accept PDP learnability as a sound non-innatist argument, then, requires accepting that the assumptions made in designing PDP experiments are not domain specific.

The idea that assumptions are either domain specific or domain independent, and that the difference is not merely one of degree or merely in the eye of the beholder -- plays an important role in discussions of language learning. It is Chomsky's belief, as well as that of many generative linguists who distinguish themselves from Chomsky, that children enter the language learning context (footnote 1) with biological constraints on the kind of grammars they will conjecture (learn). It is not an accident of particular social conditions that humans have the type of languages they have, nor a consequence of more general constraints on terrestrial communication. Human languages are the product of a specialized neuro-cognitive organ, whose development to full functionality is much like the pre-natal development to full functionality of the liver and kidneys, or the post-natal development to full functionality of flying in birds, a matter of powerful biological constraints. Change and improvement, though dependent on the environment, is strongly predetermined. The whole process is far more like a progressive tuning -- the progressive specialization of a dedicated organ -- than an enriching process where a more general purpose organ, largely non-specific, is converted by powerful learning and development processes into a computational device able to correctly assign meaning to linguistic structures.

The standard view of the PDP approach is that it represents the more general cognitive approach in which general learning mechanisms and general cognitive architectures -- ie non special purpose networks -- do the learning. Instead of interpreting language learning to be a matter of specialization of an already linguistic organ, it is more natural on the PDP model to interpret it to be the product of a progressive construction of intermediate properties which simplify the language learning problem but which might apply to domains beyond language. Networks often succeed because they build intermediate representations -- representations of properties that simplify the learning task. If these intermediate properties or representations are also found in networks learning in different domains, we have a prima facie argument that network learning of language refutes innatist views of language.

The argument must be called a prima facie argument because given the importance of what appears to be domain specific assumptions made in designing PDP experiments we may well question why we should believe that PDP language learning studies are free of domain specific constraints.

The popular reason is that the PDP design assumptions required for studying language learning are no different, in principle, from the PDP design assumptions made for studying learning in other domains. Presumably the same type of assumptions would have to be made in designing a network to learn English grammar as would have to be made if the network were to learn a function in logic, auditory perception, or motor control. They are generic assumptions. The networks are not gerrymandered or handcrafted, and the learning rule, number of repetitions, and diet are in some sense standard as well. Even if language learning requires bigger networks than those for bird song learning, or furniture categorization, the networks are just bigger versions of the same sort. Thus, runs this argument, if one day a network were in fact trained to judge English grammaticality, on that day we would have strong evidence that innate knowledge of language is not a prerequisite for language acquisition. PDP learnability of language would serve as an existence proof that specific domain knowledge is not necessary for language learning.

Now if this is a sound argument certain consequences follow that are methodologically significant. First, PDP learnability would show that poverty of the stimulus arguments about a given domain are false. The thrust of all such arguments is that certain functions are not learnable because the available data do not contain enough structure to determine the relevant function. Accordingly, such functions are deemed unlearnable by inductive methods alone: additional domain specific knowledge is required. This is the central argument generative grammarians have offered in support of their belief that `the child must come to the language learning task with inborn constraints about the possible form of linguistic rules'(footnote 2) or `with a schema of some sort as to what constitutes a possible natural language'.(footnote 3)

In overthrowing poverty of the stimulus arguments it is natural to embrace a research strategy that looks for previously unrecognized sources of linguistic information. These new sources of information may be located in the way examples are ordered in the training set, in the distribution of examples found in the set, in the frequency with which particular examples occur, or in properties of the context of usage. The methodologically salient point is that whatever the source, this extra information is available through experience. There is more structure present in the data confronting subjects than is apparent a priori. It is not surprising, then, that much PDP natural language research is devoted to uncovering the learning potential of novel sources of linguistic information. (footnote 4)

The second consequence of rejecting the need for innate knowledge of a domain is that we may substitute experiments in learnability for antecedent analysis of the domain -- at least in the first stages of research. Because a function may be learned by a PDP system whether or not we already have a comprehensive theory of the function it is not necessary to spend long hours in analysis before we set our net to learn it. One of the greatest differences between PDP approaches to language learning and innatist approaches is that innatists begin with a characterization of adult grammar and work backward to figure out how the child might arrive at this `steady state' characterization.(footnote 5) PDP and other more purely empiricist approaches work forward from the existing data about children's linguistic behaviour to some characterization of adult language. It is easy to imagine, therefore, that PDP theories of the `steady state', if such a community wide state even exists in their scheme, will be quite unlike theories of the steady state put forward in the generative tradition.

Genuine success in this methodology would mark a strong victory for bottom up research. At present, the best articulated and most widely admired method of cognitive research is the top down approach of David Marr. In this methodology formal specification and mathematical analysis take place before computational modelling. The prime defence of this top down style of research is an a priori argument: without antecedent analysis computational modelling can be no better than blind wandering in mechanism space. A priori, the chance of striking on a plausible biological design, one that might explain what we know of an organism's behavioural capacities, is simply too small to warrant attempting a search in design space undirected by prior formal analysis of the task. No general search techniques, no weak methods, can succeed. Against this negativism, the promise of PDP research is that if it can deliver a few striking empirical successes -- cases where a plausible design has been found by using a general learning rule -- we have a good reason for being optimistic that the search in mechanism space can be made tractable. The net effect might be to reset the agenda of a large, currently intransigent group of cognitive scientists.

With such weighty consequences at stake it is worth exploring carefully what PDP learnability may teach us about innate knowledge. My main concern in what follows is with the logic of the argument : vis. that a display of PDP learnability constitutes an existence proof of inductive learnability. I will use language acquisition as my focal domain because it is an area so widely discussed. But it is incidental to the main point.

It seems to me that the heart of the anti-innateness argument requires a clear understanding of what the phrases domain specific knowledge and domain independent knowledge mean. PDP learning is meant to be an example of domain independent learning -- learning that proceeds without the help of additional domain specific constraints or domain specific knowledge. If I am right the concepts of domain specific and domain independent are too ill understood to bare the weight of the innatist non-innatist rhetoric normally associated with them. Accordingly, I doubt that the agenda of most cognitive scientists will be reset by a few PDP success stories.

The paper is divided in three. In part I, I reconsider some arguments deriving from Gold's theorem purporting to show that PDP learnability could not possibly disprove the need for innate knowledge of language. Gold showed that it is impossible to learn a context-free (or more powerful) language purely on the basis of data about grammatical sentences -- a form of data that is usually called positive evidence. The learner must have access (at least tacitly) to additional information. In principle this information could come from many sources. But typically the theorem is used to justify the belief that the relevant extra information is innate and is specifically about the formal structure of language. I believe this is a mistake. But many innatists see Gold's theorem as a logical obstacle to anti-innatism -- PDP inspired or otherwise.

In part II, I begin exploring in greater depth some of the hidden complexities behind the notions of domain specific and domain independent knowledge. Part of the confusion enshrouding these ideas can be traced to the equally problematic notions of problem structure and task environment. I discuss some problems with these in Part III.

WHAT SHOULD WE LEARN FROM GOLD'S THEOREM?

In 1967 Gold posed the problem of language learning in formal terms.(footnote 6) The field of language acquisition has never been quite the same since. Gold asked the question: under what conditions is it possible to learn the correct context free grammar of a language given a set of training instances? His most significant result was that it is impossible to learn the correct language from positive examples alone. If a blind inductive program is given an infinite sequence of positive examples the program cannot determine a grammar for the correct context free language in any finite time. The data underdetermine the language. If learners are to induce correctly, they must have access (at least tacitly) to additional information.

The simplest source of this information is an informant who can tell the learner whether or not a given string is grammatical. By using these extra negative examples the program can eliminate grammars that are too general. If `negative evidence' is unavailable the language may still be learned but the additional information must come from different sources.

Gold's result is thought to be relevant to human language learning and therefore to PDP research into language learning because there is a body of literature maintaining that negative evidence is not available to children.(footnote 7) Parents do not intentionally speak ungrammatically to their children, each time pointing out that this is the way not to speak. Nor, apparently do they tell their children either directly or indirectly when the child itself is speaking ungrammatically -- not in any pervasive way. They are more concerned, it seems, with the truth or appropriateness of utterances than with grammaticality per se. But then because there is no substantial negative evidence to stop the child from choosing a grammar that generates a superset of the sentences in its mother tongue, children ought to overgeneralize wildly. They ought to be disposed to believe the grammaticality of sentences outside their language. For without additional constraints on what their mother grammar is like, children have no reason to reject sentences consistent with everything they've heard but which nonetheless lie outside their language. The psychological implication of the theorem, then, is that because children either do not overgeneralize wildly, or are able to recover from overgeneralization, they must have access to additional information about their language that has nothing to do with negative information.

Gold's theorem has often been taken as supporting innatists in their belief that the extra information about language must be inborn. (footnote 8) Part of this belief is justified on the grounds that linguistic knowledge is so specific; linguistic properties seem to resemble little else. Thus when Chomsky suggests that there are biological constraints on the kind of grammars children will conjecture he has in mind constraints on the sort of basic entities or categories -- the parts of speech -- children will consider trying out in rules of grammar. There may be analogues of such sub-recursive structures in other cognitive domains, but it is not obvious where. And when it comes to constraints on the way those entities or categories can be combined, transformed or removed, it is even less clear that there are other cognitive domains (universally learnable) which have as much structure.

To take a simple example, a child is assumed to be able to detect at an early age that its linguistic community is using subject-verb-object word order. The abstract categories of subject, object and verb are not inferred from observed regularities, it is said, they are innate. More precisely, the child is innately predisposed, at a certain stage of maturity, to represent linguistic data in structural fashion. This quite naturally simplifies the learning problem, for it allows that the input which serves as data for learning language comes in a preprocessed form. Language acquisition starts only after these abstract categories are represented by the child. They are called abstract because `their boundaries and labeling are not in general physically marked in any way; rather, they are mental constructions.' (footnote 9)

According to innatist doctrine language acquisition is further simplified by additional constraints that come into play when triggered by certain discoveries. Thus, once a child notes its language has S-V-O structure a set of triggers are fired -- or parameters set -- concerning related assumptions, such as that the language is not inflected.

Now because of all these constraints (footnote 10) on how children conjecture grammars the class of learnable grammars is an immensely reduced subset of context free grammars plus transformations. Only certain grammars are possible starting places because only certain grammars will satisfy the framework of rules and principles, and because of additional constraints only certain grammars are accessible at any point. Universal grammar, therefore, constrains the possible trajectories of learning as well as the space of learnable grammars.

Needless to say one of the most unpalatable aspects of the strong innatist position is the very specificity of the framework of rules and principles. In order to combat this view and to show that stable grammars are learnable without such specific assumptions about the nature of linguistic structures and representations, PDP oriented linguists have sought new sources of empirical information about language.

>From a PDP perspective where might this extra information come from? Two empirical sources are obvious candidates: observable facts about the communicative context, and spoonfeeding the child a special diet of sentences to learn from. Let us briefly consider each in turn.

The first conjecture is the most obvious: in early phases of language learning parents tie many of their utterances to visible circumstances. If a child were to assume that what it hears at first relates to the structure of the visual scene in front of it, then it has extra information about the content of the utterance.

No one of any linguistic persuasion, to my knowledge, has seriously denied that the context of utterance supplies valuable information to learners of a language. Ostension is an integral part of language learning. The mystery which all admit is to explain how the structuring process in visual understanding, or auditory understanding,(footnote 11) might effect the structuring process in language understanding, Indeed how are the two related at an abstract level?

One suggestion by Langacker (footnote 12) is that the child has structural schemata to help it parse visual scenes into comprehensible structures. If the structure of visual scenes is somehow mirrored at some level in the structure of the linguistic representations of those same scenes, then the child has specific information about linguistic structures that goes beyond positive examples, for it has pairings of , or, at any rate, additional information about the meaning of certain utterances.

As attractive as this suggestion is, at this stage, convincing neuropsychological details of the alleged linkage between scene parsing and linguistic parsing are absent. We suspect that visual scene parsing might be related to either syntactic or semantic structure because we currently believe that almost 50% of the brain is devoted to visual processing; that somehow vision and speech are linked since we can say what we see; and that lesions to the visual cortex can have surprising effects on speech abilities. (footnote 13) But we have no detailed accounts of how a child might use information about visual context to bootstrap its way to a rough grasp of syntax for even directly referential sentences. Moreover, assuming such accounts are one day provided, they still will not serve as proof that context plus positive instances suffice for language learning unless two other conditions are proven: 1) that a child can recognize and treat as special the communicative context without having to be taught that fact using language; and 2) that no information beyond knowledge of context is required to overcome the insufficiency of positive information alone.

In the absence of a formal proof of 2) a PDP demonstration of language learning on the basis of context and positive examples would only be suggestive in establishing their sufficiency for some languages, and some data sets. Aside from the need to undertake enough mathematical analysis to generalize the result to many languages and many naturally occurring data sets , there remains our initial concern that PDP learnability is not itself an existence proof of inductive learnability because so much information is potentially hidden in the design of the PDP experiments. PDP learnability cannot establish that no language specific knowledge is required for language learning until its own design assumptions have been shown to be language independent.

The case is no better with the second possible source of extra information -- distributional properties of positive examples, and/or the frequency with which they are repeated. If sentences are presented in a controlled manner, simple sentences being presented before harder ones, with the choice of the next sentence to be presented determined by a teacher aiming to push the student on to the best next grammar, might it not be possible to converge on an acceptable grammar?

Perhaps spoonfeeding will work. We already know that for context free grammars a careful diet of positive examples can guarantee convergence on the correct grammar. For it has been proven that for stochastic context free grammars:

> if the training instances are presented to the program repeatedly, with the frequency proportional to their probability of being in the language ... the program can estimate the probability of a given string by measuring its frequency of occurrence in the finite sample. In the limit, [this method of] stochastic presentation gives as much information as informant presentation of positive and negative examples: Ungrammatical strings have zero probability, and grammatical strings have positive probability. (footnote 14)

To date, however, this proof has not been generalized to harder than context free grammars. Eg. context sensitive,or unrestricted rewriting grammars.

When formal proof is absent empirical success is informative. A PDP network which learns a natural language when trained on a careful diet of positive examples, will, not surprisingly, be received with considerable interest. But as with claims about structure from context, experimental demonstration of language learning can at best establish the possibility of learning certain languages in certain circumstances. It is an existence proof that there are languages and data sets that can be learned by PDP networks. The trick is to show that this result generalizes to all naturally learnable languages (or that the conditions of learning English, or French are isomorphic to the structured data sets used in successful simulations); and that the assumptions built into the design and learning rule of the successful PDP system are domain independent. In short, it is necessary to show that PDP experiments in language learning do not presuppose the very assumption they wish to test: that specific knowledge of language is necessary for learning.

It is time now to turn directly to the question of what domain specificity means.

WHAT IS DOMAIN SPECIFIC KNOWLEDGE?

In AI the notion of domain specific knowledge became familiar with the development of expert systems where an explicit distinction was drawn between the general principles of reasoning built into an inference engine, and the collection of problem specific facts, goals and procedures that serve as input to the inference engine. In the simplest case, the inference engine is simply a box for deriving deductive conclusions. Domain knowledge might include premisses such as that all people are mortal, and that Socrates is a person. The output would be the conclusion that Socrates is mortal. In slightly more complex cases, domain knowledge might include premisses plus control knowledge to reduce the search of the logic engine. For given a set of axioms as input, it may take an enormous amount of undirected search of theorem space to locate the sought for conclusion. In still more complex cases, the inference engine itself might be made more powerful, capable of drawing inductive or even abductive inferences. In this last case, the engine conjectures hypotheses to explain the input data. Language learning as portrayed in the parameter setting model can be interpreted in this light if we take as the data to be explained sentences about a language, and add to that data additional inputs concerning the type and range of plausible conjectures, and interparameter constraints. See figure 1.

The AI distinction between domain specific and domain independent is not a rigorous one. The intuition appealed to is that a piece of information is domain specific if it is not useful or applicable in many different domains or many different types of problems. General strategies for deduction, induction, and abduction, then, as well as general strategies for search, sorting, classifying, normally fall on the domain independent side. On the domain dependent side we expect to find specialized search control knowledge, metrics on goodness etc, and factual data about the domain entities and their relations.

Let us see if this intuitive idea can be tightened up.

GENERAL COGNITIVE RESOURCES VS DOMAIN KNOWLEDGE.

To begin, consider why we normally suppose there is a difference between general computational or cognitive resources and domain knowledge.

Chomsky has long drawn a distinction between linguistic competence -- the system of knowledge an agent has about the grammar of its language -- and linguistic performance -- the system of linguistic behaviours an agent displays. According to the doctrine linguistic performance inevitably falls short of displaying a speaker's full competence because real agents have limited memory, calculating speed, and awareness, in short, limited general cognitive capacities. It is these resource limitations, not knowledge, which explains why we find people revealing deficits in comprehending sentences with embedded clauses and the like. Central to the competence performance distinction, then, is the idea that these deficits are general, and have nothing to do with linguistic domains in particular. Computations have costs, and these invariably become reflected in performance. Let us look at this difference between computational resource and domain knowledge more closely.

Classically, computational resources are the primary quantitative features of a computation. The amount of short and long term memory used, or the number of steps required to calculate an answer, are standard resource attributes of computations. They are measurable aspects of a process.

Knowledge, by contrast, is a qualitative feature (footnote 15) of both a computational process and a computational system. In setting up a system to perform a given computation, knowledge of the algorithm driving the computation must be installed. If this algorithm is correct the system can be interpreted as containing knowledge of this procedure as well as knowledge of certain aspects of the problem domain it was designed to work on. This latter knowledge need not be explicitly represented anywhere in the system, and indeed is usually thought to be implicit knowledge of facts about the domain that are responsible for the algorithm's success. Knowledge of the algorithm and its success conditions tend to remain constant throughout a computation. But most of the remaining knowledge in the system is explicit and tends to change moment by moment as the computation unfolds. Thus, at the outset of a problem, a system may have explicit knowledge of the input of the particular problem instance it is to solve. For example, it may know explicitly that its current problem is to derive the cube root of 125. At the close of the computation it explicitly knows that the answer is 5.(footnote 16) The trajectory of explicit knowledge states in between is a function of both resources and algorithm.

Owing to the difference in nature between resources and knowledge it is usually possible to distinguish limitations in processing capacity due to a shortage of resources from limitations due to shortages of knowledge. Shortages of resources, unlike shortages of knowledge, typically show up as a system tackles problem instances of larger size. For instance, a system endowed with the right (algorithmic) knowledge to calculate cube roots, should be able to compute the correct answer for any sized cube. But of course, as the size of the input number grows, there inevitably comes a point where either more memory is required, or more time is needed than is available. The knowledge sufficient to compute these larger numbers has not changed; so there is no need to add additional knowledge, although this would help. The problem, rather, is that the system ran out of resources.

Shortages of knowledge, unlike shortages of resources, typically show up even on the smallest problems. A system that does not know how to calculate cube roots is no more likely to hit on the correct answer for a small number than a large number. Its success is random with respect to number size. Furthermore, the addition of knowledge, unlike the addition of resources, need not improve performance in linear or even monotonic fashion. A system missing just one crucial piece of knowledge may perform no better than a system missing several pieces. Characteristically, additions to memory or computing time monotonically increase performance.

The upshot is that change in resources, seem to have domain independent effects -- either increasing or decreasing performance across domains -- while change in knowledge seems to have domain specific effects -- either increasing or decreasing performance on specific problems. This correlation becomes even more robust when we consider how a system might compensate for a loss of knowledge as compared with how it might compensate for a loss of general memory or allotted time. A reduction in memory or processing time can be accommodated on any specific problem simply by adding more assumptions -- knowledge -- about that problem's solution. As more information is made explicit about the answer set, less computation is required. This follows because at bottom computation is nothing more than the process of making explicit information available in an implicit form in a complete specification of the problem. For any particular problem, then, knowledge can compensate for resource loss. But no amount of additional computational power can make up for a knowledge poor system. If there is not enough information in a complete specification of a problem to determine an answer set, the problem is ill posed, and no amount of cleverness in search, or of brute computation can compensate. The answer is not implicit in the problem. Hence resources cannot compensate for lack of domain knowledge.

Domain knowledge, on this account, is primarily about the problem to be solved: the kinds of entities that can serve as answers to problems, their range of values, and facts about the particular problem instance. This knowledge is necessary if the system is to have a clear idea of the problem. Successful systems will have additional knowledge about potentially useful algorithms and possibly why they succeed. If the knowledge in this algorithmic component is heuristic, it concerns methods, hints and ideas that can reduce search. In principle it is not essential and its loss can be compensated for simply by generating more possible answers and testing them for correctness. To do this requires knowledge of what can serve as a candidate answer and the conditions a correct answer must satisfy. That is, essential knowledge of the problem. Accordingly, it would be more precise to say that resources cannot compensate for non-heuristic knowledge loss.

We now can operationalize at least part of the intuitive notion of domain specific knowledge as follows:

> A bit of knowledge is domain specific if its loss would have an irremediable effect on task performance. No amount of additional memory or time is able to bring performance back to its prior level.

Because this definition does not cover heuristic knowledge, which is widely understood to be knowledge of domain regularities necessary for converting weak methods to strong methods, I shall call it essential domain knowledge.

On the assumption that this operational definition captures one important aspect of our intuitive idea of domain specificity let us try applying it to the assumptions built into PDP experiments.

Recall the nature of the PDP design problem. Working from a more or less careful account of a problem -- eg. learn phrase structure grammar from a given set of positive examples -- the PDP designer must choose an appropriate network type, topology, number of hidden units, momentum factor, ordering of the data, number of trials and so forth, that he believes will succeed. To inform his choices he will make certain assumptions about the order, smoothness, regions of greatest interest etc. of the function the network is to learn (henceforth, the target function).

How are these assumptions embodied in PDP systems? The order of the target function correlates with the number of hidden units, that is, space; the smoothness of the function correlates with the number of times the data set is trained on (footnote 17) , that is, the time the leaning rule is to be run; the regions of greatest interest correlate with the distribution of samples in the data set, that is, with factors external to the computation, and the choice of net type -- feedforward, Boltzman, fully recurrent , etc. -- correlate with the type of function (associative, predictive), that is, with the structure of the network itself. In short, at least two of the assumptions built into PDP experiments -- assumptions of the order and smoothness of the target -- which on the surface appear to be domain specific, fail to be so according to our operational definition of essential domain specificity because there is a correlation between resource and knowledge.

What then are we to say about the status of these assumptions? If it is true that in PDP systems one of the ways to embody knowledge about the target function is by altering the resources available for computation, for instance, by adding (memory) units, or by adding to training time, we seem obliged to regard much of the design knowledge built into networks to be domain independent.

Admittedly, there remains the possibility that this knowledge is heuristic knowledge; it is not essential domain knowledge, but nonetheless domain specific. But I doubt that this can be correct. First, if choice of number of hidden units were important for efficiency only, and networks with the wrong number of units were capable of learning, only less likely to do so on any given learning attempt, then it ought to be possible in principle to learn arbitrary functions even in networks with few units. But we know from Minsky and Papert's analysis of perceptrons (footnote 18) that this is false. Second, if the choice of the number of learning trials were merely of heuristic value, it ought to be possible to learn functions of arbitrary smoothness. Yet as is well known, the smoothness of a function cannot be estimated reliably from noisy data. It is a desideratum which must be set. But then number of learning trials, like number of hidden units, is not merely heuristic knowledge, it is essential knowledge, for it effects the very way we understand the problem.

Should we reject our operational definition of essential domain knowledge, or should we reject the idea I have been tacitly assuming all along, that choice of hidden units and trial repetitions is domain dependent, that is, domain specific knowledge? My inclination is to drop the definition. In fields like econometrics where statistical estimation of target functions is the stuff of life, the shape of the target (eg. y=ax3 + bx2 + cx + d or yt = ayt - 1 + b) drawn from the theory of economics. The econometrician `relies heavily on a priori knowledge [drawn from] economic theory'. (footnote 19) These assumptions are not merely heuristic; they are necessary to an adequate specification of the estimation problem. But then are they not as domain specific as assumptions can be? If domain specific knowledge is necessary for statistical estimation of functions in econometrics, why would it not also be necessary for PDP modelling of cognitive capacities, which is also interpreted as a mechanism for estimating functions?

Let us try another tack at making more precise the intuitive notion of domain specific knowledge.

TRANSPARENCY OF DOMAIN KNOWLEDGE

Why do the assumptions made in the language learning models of generative linguistics seem to be domain specific? One easy answer is that those assumptions transparently refer to entities, facts and regularities of languages.

Parameter setting models are based on the theory of UG (universal grammar) which adverts to structural descriptions of sentences, to constraints on transformations between those essentially linguistic structures, and to entities or notions such as bound anaphor which are undefined outside of language studies. Parameter setting models are transparently about language because the concepts mentioned in these language learning models cannot be readily divorced from language. One could define a set of mathematical structures that are isomorphic to the structures discussed in generative linguistics. And so convert linguistics into a branch of mathematics that now is about formal structures rather than human languages. But these formal structures are not motivated by extra-linguistic considerations. They are solely motivated by the study of language. Thus it is not an accident that there is an independent mathematical theory of tree structures, but not of phrase structures, or bound anaphors. These last are too idiosyncratic. See figure 2.

It is worth putting this argument in simpler terms. What makes a set of assumptions specific to a domain is that those assumptions are about entities and structures that are special to that domain. They are not general mathematical entities, such as functions or graphs, which have general application to many fields. They are highly specific and idiosyncratic -- so idiosyncratic that the only natural way of talking about those entities and structures is in the terms developed in the empirical domain they belong to. Non-generality of structure naturally leads to transparency of discourse.

> Knowledge is domain specific if it transparently refers to entities and facts that are not general or generic, but rather specialized and idiosyncratic to the domain in question.

On this account PDP based theories of language learning, based as they are on assumptions about the form, style and size of networks needed to instantiate certain linguistic functions, the learning rule, the kind and distribution of data it will be trained on, and the number of times the data will be sent through, mention nothing that is transparently about language. Virtually the same assumptions could apply, for all we know, to auditory processing, linguistic processing or visual processing; and the very same network and learning rule if fed different data could be used to learn other functions. So prima facie language learning networks do not contain knowledge about the linguistic domain per se; they contain knowledge about the formal properties of certain functions. Hence PDP learning models contain no domain specific knowledge.

As reasonable as this argument may seem there is at least one good reason for not accepting it: descriptions do not have to appear to be about the objects they refer to to actually refer to them. Transparency of reference cannot be necessary for domain specificity.

The argument for non-transparency is familiar in philosophical circles. Descriptions may be referentially opaque. It is possible to refer to the actions of a pocket calculator as the manipulation of numbers rather than the manipulation of numerals or electric currents, and to the field of physics as whatever physicists study. The common feature of these descriptions is that they refer indirectly. They seem to be about one thing -- numerals, electric current, the actions of physicists -- but in fact refer to entities that are more directly designated by other expressions -- numbers, quarks and force fields.

But then we can grant that transparency can serve as a sufficient condition for knowledge being specifically about a domain, yet deny that it is a necessary condition. It is entirely natural that descriptions of networks and data sets appear to be about networks and data sets, and that the assumptions going into the choice of an architecture seem to be about the order and shape of the target function, yet they nonetheless refer to assumptions about linguistic properties and structures. Transparency is not necessary.

LAW-LIKE ATTUNEMENT TO DOMAIN REGULARITIES

Perhaps the strongest reason for considering parameter setting models of language acquisition to be so clearly about language specific entities and facts is that every accessible parameter setting in one of these theories defines a possible language -- a possible human grammar. Parametric space somehow mirrors linguistic space. The intuition here is that the parametric framework is perfectly tuned to the structure of human language. (footnote 20) This means that the assumptions that are built into a parametric model are not just about English or French or a few other natural languages -- ie. particular examples of the language learning task. They are about any language that a human now or in the future could speak -- any example of the task. All and only possible human languages are definable as vectors in parameter space. No non-human languages are describable. See figure 3. Thus what makes parameter setting models seem to be about human language rather than, say, about some formal game, is that they are tuned to the possible, not merely the actual. The formalism of parametric theories is (supposed to be) perfectly adapted to language. It is related in a lawlike way to language because it captures what is essential to language -- the constraints on possibility. See Figure 3.

The idea here is that the way to decide whether a system has knowledge about a given domain and not about some other domain is to consider the counterfactual implications of the assumptions it embodies. There is a familiar precedent for this. The normal way of deciding whether a person has a particular concept -- say the concept of cup -- is to see if he or she calls all cups cups and then to see if s/he is disposed to go on to use cup in the right way in the future. Shown cup-like objects they have never before seen they must classify them the way people who we agree understand the term would also classify them. That is, we assume they have the right counterfactual dispositions. It is this counterfactual ability that is thought to distinguish coincidental connection from lawlike connection. It locks the concept to its referent.

We can state this condition on domain specificity as follows:

> Knowledge is specific to a domain if it is connected in a lawlike way to the possible entities and structures of that domain.

Although we cannot use this as an operational definition of domain specific knowledge unless we can decide when the elements of knowledge are connected to entities in a lawlike way, we can still put to use the idea that assumptions built into a computational system are domain specific, or task specific, when they are exactly tuned to the properties of the task.

For instance we can ask what conditions would a network have to satisfy to be counterfactually attuned to language in just the way parameter setting models are. If we were to discover that successful language learning networks satisfy these conditions, then we would have reason to suspect that the assumptions that go into their design are equal in size and specificity to those built into parameter setting models. If we think the one has domain knowledge built into it, we ought to believe the other has it too.

Here then are the conditions on a networkese version of a parameter setting model.

> 1) There is a well defined family of networks N0 -- the class of networks pre-tuned to the structure of human languages -- that have the appropriate design to learn any human language when subjected to the same type of linguistic data as human children.
>
> 2) The trajectory of grammars (system of linguistic behaviours) these networks would describe as they converge on the steady state grammar mirrors that of human children. That is, when learning human languages, these networks are constrained to pass through phases or stages of behaviour that duplicate those which children pass through. Only certain grammars can be tried out in the course of learning. The learning rule, therefore, must be such that when coupled with the data set it issues in `stable points' -- regions of current best estimate of the best function fitting the data -- that mimic allowable vector trajectories in parameter space. Each of these stable points represents one of the possible grammars the child is trying out. It is a grammar of a possible natural language.

If the choice of architecture, learning rule, diet, number of epoques and the rest are as constraining to network and network trajectory as 1) and 2) I cannot see how anyone can deny that network models of language contain domain specific information, and that N0 , in particular, has as much information about language as a parameter setting model. That would settle the question once and for all whether PDP networks have domain specific knowledge in them.

Once more, however, the matter is not so easily resolved. There is at least one good reason for supposing that the assumptions that go into choice of architecture, etc., are not in fact this constraining. Gradient descent methods, such as backprop, are too sensitive to initial settings of the weight vector to expect all paths leading to stable grammars to be similar. The same network starting from slightly different intializations could describe substantially different trajectories. The same is true if we are comparing the trajectory of different networks in N0: each will have its own idiosyncratic path from initial to final state. Moreover, gradient descent methods are weak methods; there is no provision for extra control information (footnote 21) of the sort that would overrule choice of the steepest descent. As a result, there is nothing to prevent networks from trying out weight vectors that have no counterpart in parameter space. They are not prohibited from temporarily settling on intermediate representations and sub functions in their inductive search for the steady state grammar just because those representations or sub functions are not linguistically `natural'. From the network's vantage nothing is linguistically natural or unnatural. The learning rule is domain independent.

Here again is an argument for less innate domain knowledge. But note, it cannot be an argument for no domain knowledge. For in the phrase `counterpart in parameter space' we are making tacit reference to an interpretation function that maps vectors in weight space to expressions in another more linguistically transparent formalism. If we could agree on such a formalism we could apply it to the initial conditions of the entire family of successful PDP language learning networks and look for invariants. Accordingly, in my opinion, the interesting question PDP studies of language learning raise is not how much of language is innate, but what about language is innate.

To solve this will require agreeing on an interpretation function for language learning networks. One major source of dispute among PDP oriented linguists and generative linguists is over what the appropriate linguistically transparent formalism should be. It is fairly clear that some such formalism is necessary. For if there were not some way of interpreting the linguistic information in networks there would be no way of knowing whether two networks converge on the same grammar or different grammars. Similarly there would be no way of knowing if there were any interesting linguistic information present in the starting state of all successful networks. It would not even be possible to derive linguistic generalizations from studying families of successful networks. So settling on an interpretation function is essential to PDP linguistic studies. But it also throws us right back to the question of what constitutes the domain of language -- a question which some see as the defining question of the empirical field of linguistics.

SUMMARY

I have been considering some of the problems undermining efforts to use PDP simulations of language learning as existence proofs that innate knowledge of language is not necessary for language learning. Virtually all parties to the dispute agree that some knowledge or some learning strategies must be innate but there has been widespread disagreement over how domain specific that innate knowledge must be.

I tried to elucidate the notion of domain specificity by appealing to reasonable intuitions we have. We think that there is a genuine difference between cognitive limitations brought on by scarce cognitive resources and cognitive limitations due to insufficient knowledge. A difference, moreover, that might clarify the meaning of domain specific. But when applied to PDP style architectures this distinction proved parochial.

I then tried linking domain specificity to referential transparency: an assumption is about a specific domain if the entities and structures it refers to are idiosyncratic -- highly specialized. The more specific the entities the fewer the domains those entities could belong to. Assumptions about those entities, therefore, would have to be about the specific domain they belong to. This intuition I granted could serve as a sufficient condition for domain specific knowledge, but it was too exclusive to be a necessary condition. PDP systems might be built on more generic assumptions about functions, and so forth, and yet incorporate domain specific knowledge.

This led me to my final intuition that an assumption that is built into a system carries information specific to a domain if it is connected to entities in that domain in a law-like manner. This has the virtue that some assumptions can be about non-idiosyncratic entities. But it left us grasping for a way of translating the assumptions built into a computational system into a transparent formalism. I argued that because networks are not transparently about language we must have an interpretation function to map PDP design assumptions into expressions in another more linguistically transparent formalism. Else we could not determine what entities particular system assumptions corresponded to. The very question of linguistics, however, is what should this formalism be. It is the hope of PDP linguists that the way to discover this formalism is by extensive PDP modelling. It is too early to say how successful this approach will be. One thing we can be certain of, though, whatever theory is eventually preferred it will show that there is substantial information about language in the initial states of language learning networks. What I hope I have established is that this is not in itself an interesting question. The real question is what is this innate knowledge of language.

I want to close now with an argument that should chasten anyone who believes that vanilla domain assumptions will suffice for PDP learnability of language, and that the vaunted power of PDP systems to learn intermediate representations can do away with all but the most rudimentary assumptions about language. In my opinion it is more likely that substantial innate knowledge of language -- in particular, knowledge of the constraints on intermediate representations -- will have to be built into PDP language learning systems, although as yet we have no settled idea what this innate knowledge will look like and how it will play itself out in the design of networks complex enough to learn natural languages.

THE NEED FOR CONSTRAINTS ON INTERMEDIATE REPRESENTATIONS

In any multi-layered PDP system part of the job of intermediate layers is to convert input into a suitable set of intermediate representations to simplify the problem enough to make it solvable. One reason PDP modelling is popular is because nets are supposed to learn intermediate representations. They do this by becoming attuned to regularities in the input.

What if the regularities they need to be attuned to are not in the input? Or rather, what if so little of a regularity is present in the data that for all intents and purposes it would be totally serendipitous to strike upon it? It seems to me that such a demonstration would constitute a form of the poverty of stimulus argument.

The example I wish to discuss is illustrative only. I have no reason to suppose that it is especially analogous to the problem of language learning. But it is consistent with the theoretical nature of much of generative linguistics.

Consider, then, the problem of representation posed by the mutilated checkerboard. See figure 4. The problem is a straightforward tiling question: can dominoes 1 by 2 in size, be placed so as to completely cover an 8 by 8 surface with 1 by 1 regions missing from position (1 8) and (8 1).

To solve tiling problems in general requires substantial search. But as is well known, we are able to quickly solve this particular problem by treating the surface as a square checkerboard missing the opposite ends of a diagonal. We can then exploit the familiar property that all tiles along a diagonal of an n by n checkerboard will be the same colour. Clipping the ends off a diagonal will therefore reduce the number of, say, black squares by 2 while leaving the number of white squares constant. Because each domino covers exactly one black and one white square there can be no pattern of tiling to completely cover diagonally mutilated boards. See Figure 4.

There are several ways we might interpret this patterned Euclidean space but the one I prefer treats checkering as akin to a geometric construction. A legitimate geometric construction never violates the rules of geometry. It adds additional structures which if well chosen alter the original problem situation by making explicit properties and constraints that were otherwise implicit. When such properties are felicitous they make discovery of the target property easier.

In checkering a board we are adding a structure to the bare statement of the tiling problem. This structure is not in the input, so it is not inductively inferable. It is a legitimate addition because the way a given space will checker, and the set of properties that follow from checkering it, is determined by the axioms of the space. But there are also an indefinite number of structures consistent with Euclidean geometry which we are not considering, because they are irrelevant to solving the current problem. Choosing the right structure to add requires insight. Accordingly, we ought to view checkering to be a hint, or better, a facilitating property, that lets us discover properties of Euclidean surfaces that would otherwise be hidden.

What if the discovery of grammar requires the same felicitous addition of structure to the data of discourse? If such structure is consistent with the data but not inductively derivable from it then inductive engines, such as PDP systems, might yet discover grammar by other more lengthy methods, but miss the quick discovery that comes from operating with the right hint. This is the spirit in which I interpret Chomsky's arguments about the necessity of recoding the input of speech in structured form.

Now prima facie there is no reason PDP networks cannot be designed to bias recoding input in ways which lend themselves to discovery of the best intermediate representations. But to do so requires substantial prior analysis of the linguistic domain. The translation to networkese may be as natural as constructing a net in phases, with the global language learning problem broken down into tractable subproblems, each assigned to separate nets to learn. Or again, perhaps the solution will involve creating low bandwidth linkages between appropriately designed subnets. If either of these cases are close to the mark, PDP theorists will have to enter the design phase with a tremendous amount of domain specific information. For now we are not just concerned with the order of a function but with its internal structure too. That is, we have decomposed the function into a set of composable parts -- each with its own order etc -- and we have chosen a way for the parts to interact.

ACKNOWLEDGEMENTS: This essay arose out of reflections on the Interdisciplinary graduate seminar in Cognitive Science at UCSD on Language Acquisition. Special thanks to Farrell Ackerman, Marta Kutas, Marty Sereno, David Shanks, and Mark St John for helpful conversations on the topic, and to Liz Bates for animus. Funding for this research came from NSF grant DIR89-11462.

[1] Strictly speaking the language learning context is entered only after having solved the bootstrapping problem See Pinker, S. 1987. The bootstrapping problem in language acquisition. In B. MacWhinney, ed. Mechanisms of Language Acquisition. Hillsdale, NJ: Erlbaum.

[2] Pinker, S. 1989. Language Acquisition. In M. Posner, ed. Foundations of Cognitive Science. Cambridge, MA: MIT Press. p 370.

[3] Wexler K. and P. Cullicover. Formal Principles of Language Acquisition. Cambridge, MA: MIT Press. p 4.

[4] See Elman J. 1991. Incremental Learning, or the Importance of Starting Small, Proceedings of 13th Annual Conference of the Cognitive Science Society, 1991 Erlbaum, pp443-448, for an example of PDP research dedicated to uncovering new sources of linguistic information. Elman suggests that children may succeed in simplifying their linguistic problem by searching, at first, for grammaticality in restricted word sequences. This restriction is meant to correspond to the child's limited attention span which prevents retention of more than a few words of a sentence at a time. As a child's memory and attention grows it is able to bootstrap to more realistic grammars.

[5] Chomsky put the matter this way.

> . . . we begin by determining certain properties of the attained linguistic competence, the attained steady state Ss. We ask how these properties develop on the basis of an interplay of experience and genetic endowment.

Chomsky, N, `On Cognitive Structures and their Development, A reply to Jean Piaget', in M. Piatelli-Palmerini, ed. Language Learning: A debate between Noam Chomsky and Jean Piaget. Cambridge MA: Harvard p 48.

6 Gold, E. 1967. Language identification in the limit. Information and Control 10:447-474. Gold's theorem can be established only if we are explicit about:

> 1. the space of possible languages, and the one which is the target;
>
> 2. the type, order and frequency of information available to the learner which is relevant to determining the correct language;
>
> 3. the learning strategy that tells the learner how to create and change its hypothesis about the target on the basis of data from the environment; and
>
> 4. a success criterion for deciding if the learner has conjectured the target.

Needless to say when any one of these assumptions are made specific they may not resemble the true situation facing natural language learners.

7. The mich cited original work in this area is Brown R., and C.Hanlon 1970. Derivational complexity and the order of acquisition in child speech. In J. R. Hayes, ed. Cognition and the Development of Language. New York: Wiley.

8. It is worth noting that Chomsky himself does not appeal uniquely to Gold's theorem. I have argued that we can, under an appropriate idealization, think of the language learner as being supplied with a sample of well-formed sentences and (perhaps) a sample of ill-formed sentences -- namely, corrections of the learner's mistakes. No doubt much more information is available, and may be necessary for language learning, although little is known about this matter. [Chomsky `Discussion of Putnam's Comments' in op cit Piatelli-Palmerini p 312.]

9. Chomsky `On Cognitive Structures and Their Development: A reply

to Jean Piaget' in op cit Piattelli-Palmerini, p 39

10. The Specified Subject Condition -- SSC -- is a more complex example which shows the type of innate constraints Chomsky has in mind that might operate on transformations. The SSC asserts roughly that no rule can apply to X and Y in structures of the form ...X... [...Y...]... where X and Y are noun phrases and [...Y...] is an embedded sentence or noun phrase, if the embedded phrase contains a subject distinct from Y. Under normal conditions the pairs each of the men ... the others and the men ... each other are interchangeable without substantial change of meaning. For example,

(1) Each of the men likes the other.
(2) The men like each other.

But in some contexts this not true. Sentence (3) ought to transform to (4). But (4) is neither

(3) Each of the men expects [John to like the others].
(4) The men expect [John to like each other].

synonymous with (3) or even a well-formed sentence of English. The reason the transformation is blocked is that the embedded sentence in (4) contains a subject John which is distinct from each other so that the relation between X and Y is blocked by SSC.

11 See Bregman, A. 1990. Auditory Scene Analysis, Cambridge MA: MIT Press.

12 See for instance Langacker, R. 1986. Foundations of Cognitive Grammar, vol 1. Stanford CA: Stanford University Press, and Lackoff, G.1987. Women, Fire, and Dangerous Things. Chicago IL: University of Chicago Press.

13 See Rubens, A. B. and A. Kertesz. 1983. The localization of lesions in transcortical aphasias. In A. Kertesz, ed. Localization in Neurosychology. Academic Press, pp 245-268. Also see Sereno, M. I. 1991. Language and the Primate Brain. Proceedings Cognitive Science Society , Hillsdale NJ: Erlbaum. pp 79--84.

[14] Clarkson, K. 1982.Grammatical Inference, in Cohen P. and E. Feigenbaum, eds. The handbook of Artificial Intelligence. Vol 3. p 500.

[15] The distinction between knowledge as a qualitative property rather than a quantitative one does not mean that there cannot be more knowledge or less knowledge built into a system. It does mean, though, that we cannot measure exactly how much using a familiar quantitative scale. This restriction applies because first knowledge is an attitude to propositions, and propositions are notoriously difficult to measure. Second, what a system is thought to know can vary with context and indeed with what aspect of system behaviour we are studying. [16] For a preliminary discussion of this idea, see Kirsh D. 1990. When is information explicitly represented?, in P. Hanson, ed. Information, Language and Cognition. Vancouver BC: UBC Press.

[17] Both the updating rule and the momentum associated with movement in weight space can also effect smoothness.

[18] Minsky M, and S. Papert, 1988. Perceptrons (2nd ed.). Cambridge MA: MIT Press.

[19] M. Dutta, Econometric Methods, South Western, 1975, p 10.

[20] It is not clear that circularity can be avoided here. For if the defining feature of a humanly learnable language is that it is consistent with Universal Grammar (UG), and the meaning of UG is that it defines the space of humanly learnable languages -- the innate restrictions imposed by the language organ on what languages humans might possibly learn -- then it is analytically true that UG is perfectly tuned to the structure of human languages. This is one way of guaranteeing a necessary relation between UG and the domain of language.

[21] This is not literally true. Most backprop methods allow for a momentum parameter whose job is precisely to slow the jerkiness of gradient descent. That is, to prevent taking very short steps downhill that go off in a different direction than one has been moving, an extra input is added to make smooth transitions more desirable. But the point still stands that this is not a flexible control method that allows backprop to make use of linguistic information in its moment by moment choice of how to

update weight vectors.

[CRL Newsletter Home Page]
[CRL Home Page]

Center for Research in Language
CRL Newsletter
Article #