NOISE AND KNOWLEDGE ACQUISITION

Michel V. MANAGO, Yves KODRATOFF

Inference and Learning Group
LABORATOIRE DE RECHERCHE EN INFORMATIQUE (U.A. 410 du CNRS)
Bat 490, Universite de Paris-sud, 91405 ORSAY Cedex, FRANCE

Abstract:

In this paper we analyse how noise can affect Knowledge Acquisition from a Machine Learning perspective. We present some methods to detect and treat noise that goes beyond modulating numerical coefficients and show that noise cannot be viewed as a single entity. There are several different types of noise and noise is not only wrong information.

I. INTRODUCTION

Formalizing the knowledge needed to solve a real world problem is far from being a trivial task. As noted in (Clancey, 1986) (Mc Dermott, 1986), Knowledge Acquisition (KA) is not the process of transfering a mental model that lies somewhere within the brain of a human expert, but the familiar scientific and engineering problem of formalizing a domain for the first time.

The classical methods to achieve this result is based on the domain expert's ability to explain his (or her) behavior:
- a knowledge engineer interview the expert and attempts to formalize his knowledge
- the expert himself is trained to construct a computable model that is extentionally equivalent to his own model

This is usually a cumbersome process because human experts are trained to solve a task and not to explain how they obtained the solution. Furthermore, the final model frequently contain bugs and is only representative of that specific expert (two different experts almost never agree on what should go in the knowledge base). This causes several problems in particular those connected with the system's maintenance.

An alternative to the traditionnal methods is to use Machine Learning (ML) tools. From a set of examples of expertise, the learning system automatically constructs the rules. Several experiments have shown that very good results can be obtained this way (see for instance (Michalski & Chilauski, 1980) (Quintan, 1986)). These techniques partially solve the problem of getting the expert to formalize his rules. However, the expert still plays a critical role since he must provide the vocabulary needed to describe the events (the language of description), the domain specific properties (valid axioms and constraints, default and common sense knowledge and so on), and that he must validate (or even provide) the set of training examples.

In a real application, the data given to the system by the expert (whether it is obtained manually or automatically) contains noise (wrong information, lack of information, unreliable information). We view noise as a critical problem that must be solved to build an accurate Knowledge Base.

The traditional approach to handling noise consist in attaching numerical coefficients such as certainty factors (Buchanan & Shortlife 84) to the rules. However, we feel that this approach does not solve all the problems relating to noise and that it generates some new ones (for example, it is difficult for a human expert to relate to a large set of such rules and to generate or validate these).

In this paper, we examin different types of noise, which affect KA at different steps and which yield different procedures of detection and treatment.. Some methods developed here are common to both traditional and automatic KA techniques (for example cross-examination of expertise) but we actually studied how noise affects KA from a ML perspective. The ML tools that are used for this research are the decision-tree maker NEDDIE (Corlett, 1983) and the generalizer MAGGY (Manago, 1986). These are respectively descendants of the algorithms ID3 (Quinlan, 1983) and AGAPE (Kodratoff & Ganascia, 1986).

The material presented in this paper is based on a research project which aims at automatically building a Knowledge Base to diagnose tomato plant diseases and to compare the result with an existing Knowledge Base that contains over 350 rules (this work started in October 1985 and is done in collaboration with the British Research Laboratory of the General Electric Company, the French company Cognitech and the French "Institut National de Recherche en Agronomie"). The example base contains observed examples and expert generated examples.

This research has been partially supported by the European Economic Community (ESPRIT-AIP contract P1063, the INSTIL project) and by the GRECO and PRC "Intelligence Artificielle". The other partners of INSTIL are GEC research (UK) and Cognitech (F).

We are currently studying another large scale application, air traffic control, and the preliminary results brought a few contributions to this paper. A case library is currently being built (work started in October 1986) in collaboration with the "Centre d'Etudes de la Navigation A6rienne": by looking at a radar picture, the human expert describes what he sees and explains his behavior. Some of the material presented here was and is still being tested on a smaller application in law. The examples come from the outcome of 57 cases in a court of law, that describe the legal actions and the duties of the mayor of a city (work started in fall 1986).

II. GETTING THE INITIAL KNOWLEDGE

Knowledge located at different levels must be obtained from the human experts (Alexander et al, 1986). In this paper, we have divided the KA process into the following three steps:

1) Expert Interaction + Literature on the domain
2) Concept formation from examples (learning full descriptions of the high/intermediary level concepts)
3) Rule learning (learning diagnostic rules and meta-knowledge)

The last two use the ML tools.The reasons for going through the second stage will be explained in section II.A.2. Most of the methods that we will describe for detecting and treating noise rely heavily on the ability to go back and forth between the first and second components.

Note that we call concept a function f that partition the space of instances into the instances that verify the function (the set of instances whose image is TRUE) and those that do not. A low level concept is the basic building block *(for instance the symptoms),* a high level concept is what we are trying to learn *(for example the diseases)* and an intermediary level concept is something in between that is "interesting" *(for example, the diseases caused by a fungus).* It is important to emphasize that there is an intentional component in a concept (its purpose).

We will not describe in details the second step (see (Kodratoff & al, 1986) (Manago & Kodratoff, 1987)). The basic idea is to grow a decision tree with NEDDIE and to generalize the clusters with MAGGY to obtain full descriptions of the concepts. The resulting system is then model-driven which makes it more robust with respect to noise (Fu, 1985).

II.A. INTERACTION WITH THE EXPERTS

II.A.1. Getting the Initial Vocabulary

There is no universal method to obtain the initial vocabulary. The descriptors can be collected bv searching the literature, by interviewing an expert (Regoczei & Plantinga, 1986) and so on. In our application (plant pathology) we started with an existing expert system which was built by an expert and we asked another expert to filter off the irrelevant (noisy) descriptors. *A descriptor like BIRD-EYE-SPOT was converted into "two round spots, the darkest one in the center of the other.*

Cross-examination between several sources of knowledge for fighting noise will come up several times in this paper. This is similar to a technique used in picture analysis (computer vision, signal processing) to filter noisy pixels. Two pictures are taken and AND-ed together to remove false pixels or OR-ed together to fill missing pixels. We take a "picture" of the domain with an expert and compare it with "pictures" taken by other experts and/or by the persons who will use the expert system.

The choice of descriptors depends on the underlying logic used by the system. To represent the examples and the rules we use a unit-based first order logic language (Nilsson 1984, Chapter 9.1). First order logic enables using low-level concepts such as RED and FRUIT. Intermediate concepts such as FRUIT(x) & RED(x) will not be represented as RED-FRUIT(x) unless it is "interesting" (symbolizing switchover points (Fu & Buchanan, 1985) or intermediary subgoals (Kodratoff et al, 1986)). As noted in (Clancey, 1983) "intermediate knowledge provides better explanations capabilities for the system and thereby increase the understandability for the user. Nevertheless, we emphasize the use of low-level concepts to enable communication and cross-examination between different experts. It also removes some of the initial bias introduced by the expert who provided the language of description (Utgoff, 1986).

The choice of what is a constant and what is a predicate is usually independant of the application. When there is only one person named MARY in the universe, MARY can be a constant. If there are several persons. The choice of the low-level concepts (or primitives) varies from one application to the other even in the same domain. *For example, in the world of blocks, if the application is to move blocks around as in SHRDLU (Winograd, 1972), CUBE is a valid primitive. On the other hand, if in the system is to perform analysis of visual scenes in that same world of blocks the concept of CUBE will be described in term of concave/convex interior/boundary lines (Waltz , 1975) etc...* Identifying low-level concepts can be fairly difficult *(in our ai-traffic control application the expert has a hard time describing the knowledge contained in the radar picture even in natural language).*

The initial choice of primitives to represent knowledge is necessarily ad hoc (Schank & Carbonell, 1979). Thus the choice of descriptors is never considered as final and is modified depending on the results. We are very sensitive to the importance of a good set of descriptors and its relation to noisy data. A good choice of descriptors enables all things to be said cleanly (Hayes, 1984) and lot of problems connected to noise can be solved by improving the language of description. In these cases, using numeric uncertainty is not appropriate.

II.A.2. Object-Level Knowledge

During stept we also obtain some domain specific knowledge (a model). This is:

- generic objects (all fruits have a color, a size, a texture etc...)
- default properties (leaves are normally green)
- relations between objects that always hold (a leaf is part of the plant, yellow is a light color)
- axioms (if there are a lot of spots on a leaf, then the size of the spots must be small)

There are various ways to represent background knowledge. We have chosen to encode it using a frame-based language. For example, the frame-unit LEAVES has a slot COLOR filled with ($DEFAULT (GREEN)). This property may be inherited by a specific instance of that frame (or an ISA descendant) when information is missing (noise). Slot fillers may be simple attributes or procedural calls (deamons). For example, the COLOR slot of the TOMATO-FRUIT frame-unit is filled with an $IF-NEEDED deamon:
-If the fruit is young, the procedure returns green
• if it is mature it returns red.
We have implemented all axioms as deamons to deduce missing low-level concepts (noise). Axioms could also be implemented as constraints (if the color of the fruit is not red, then the fruit is not mature) to detect wrong information (noise) when a constraint is violated. We have not yet implemented this in the system.

Then we must obtain the training example base. This is achieved by filling up questionnaires. The information contained in these can then be generalized to produce characteristic descriptions (full descriptions) of the concepts.

II.A.3. Using characteristic descriptions

In terms of Version Space (Mitchell, 1985), a consultation rule is a generalization in the G set: or set of most general consistent generalizations. A lull description is a generalization in the S set or the set of most specific generalizations that cover all the positive examples. In perfect domains, the positive examples exactly represent the sufficient conditions for belonging to a concept and the negative examples the necessary conditions. Thus, the G and S sets meet when all examples are processed. As noted in (Wittgenstein, 1953) (Fu & Buchanan, 1985), this almost never occur with natural concepts. As a consequence, there is a "gap" , the Version Space is not empty. We view the lack of training examples as a very important kind of noise in ML

High-level concepts are not merely descriptions of the information provided by the training examples but they also have a predictive power for classifying unseen events. When the unseen event is less general than the complete description (for instance, the event "yellow spot on leaves" is less general than "symptom on leaves") then using full descriptions or consultation rules will yield the same results. When the unseen event is in the Version Space (i.e. more general than the full descriptions and less general than the consultation rules) then by using full descriptions the system will not be able to classify it , but if it uses consultation rules it will. This is not a positive feature since we do not have any information on what the class of that event might be. There can even be conflicting consultation rules that all claim the event.

A solution is to use full descriptions (or use consultation rules and check if the answer matches a full description) and to learn incrementally when an event is not recognized (as done in the Version Space method). The expert is asked afterward for the class of the event and the system updates its set of full descriptions in order to recognize similar events in the future. Note that on the long run the result will not be as good as the ones obtained by processing the example base in one shot and will depend on the order in which the events are presented.

Another problem for ML tools with generalizing consultation rules and/or cross-examination of such rules, is that they implicitly contain strategies in the form of intermediary subgoals. Consider the following example where the system recognizes vehicles such as CARS, MOTORCYCLES and BICYCLES.

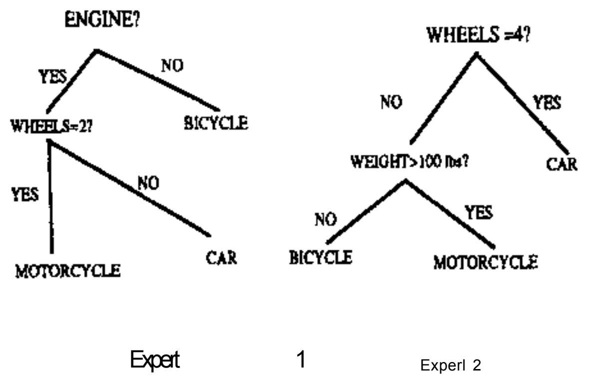

Fig. 1: Two decision trees to classify vehicles

Two experts build the decision tress of fig. 1 or give the equivalent production rules representation (Corlet, 1983) (Quinlqn, 1986). The two experts have different strategies. One prefer to first find if the vehicle belongs to the class "vehicle with an engine" and the other to the class "vehicle with two wheels". There are various reasons why their strategy may differ (for instance one of them could live in Pekin where there are a lot more bicycles than cars). Both sets of rules are correct and usable by an expert system but from a learning point of view, there are no relevant similarities among these.

The problem with consultation rules, is that they contain a single explanation (chain of reasoning) of why an element belongs to a certain class. Since there can be several other valid explanations, two experts may come up with different ones. If the system is to learn from rules given by different experts, each one must provide full descriptions of the concept. Intermediate knowledge and strategies are then represented explicitely.

For instance, a full description of the concept car, is "a colored vehicle with four wheels, which weights between 1000 lbs and 4000 lbs, which has a windshield, an engine, a rear view mirror, a hood and soon

Full description contain all the information that is common to the positive examples (or to the clusters of positive examlples) but they are generalized descriptions. The system does not need to memorize all the examples encountered as it is the case in the exemplar theory (Smith & Medin, 1981).

From the set of full descriptions, it is possible to dynamically grow alternative decision trees when a user is not able to answer a question (noise). For example, what if the user does not know whether or not there was an engine in the first decision tree? The system could then grow the second tree an find another path leading to the solution.

The usual way of dealing with the problems mentioned above is to attach certainty factors to each characteristics. For example, we would have rules such as DOOR IMPLY CAR (OF .76), TRUCK (CF 4) and so on. However this is not always the optimal solution considering that:

- We have to find methods to combine these rules. This is not easy when the independance assumption does not hold (when low level concepts can be related). For example, if the rule DOOR-HANDLE IMPLY CAR (CF .76), TRUCK (CF .4) is fired after the previous one, it sould not modify the state of the consultation.
- When an unseen event is located in the Version Space, the result of the consulation is as meaningless as if the system would have used flat consultation rules. However, since there is some uncertainty associated with the conclusion, the problem is hidden.
- When the statistical information brought by the example base is irrelevant (as it is the case when we ask for experts full descriptions) the CF cannot be properly computed. Even with a true case library, the statistical data can vary over time or changed when moved in a different environment.

We now study how noise affects the training examples and the full descriptions of the concepts.

III. NOISE IN A RULE BASE

Noise is present in a knowledge base when it does not truly reflect the environment we want to learn from.We define noise as being:

- Erroneous information
- Missing information and bad language of description
- Unreliable data

There are different sources of noise, different effects of noise and different kinds of noise. Hence, it is rather difficult to speak about noise as a single entity. Note that we are not concerned with noise in the expert's strategies (expert wondering off on the wrong track) since we are interested in full descriptions.

There are two separated issues which are:
- forming concepts (learning full descriptions) when the training data contains noise
- generating high-level concepts that are robust with respect to noise.

III.A. Unreliable information

In III.A., we assume in that the examples are correct and that the noise does not originate from the language of description. Unreliable data is noise that naturally originates from the low-level concepts themselves. We have identified several kinds:

1.a) Concept Is hard to see.
For instance, a discriminating feature of "Colletotrichum coccodes" is the presence of tiny black marks on the roots, less than one millimiter in size which are hard to see and often missed during consultation.

1.b) Concept polymorphy (symbolic noise)
This happen when two low-level concepts have a non-empty intersection and may be confused. For example, when does dark grey stops being grey to become black?

1c) Concept requires a high skill of expertise to be Identified
While an expert is not likely to confuse a brown spot with a rotting tissue, an unskilled user is. Note that experts usually know when to carefully look for noisy concepts of type 1.a or 1.b and that noise in these categories often falls into category 1 .c.

We can detect noisy concepts of the type 1 .a, 1 .b by cross-examination of information given by several experts. When experts disagree on the value to give to a specific descriptor, then the associated concept is probably unreliable. Likewise, if the experts and the users disagree, then the concepts probably require skill to be identified. This is why we ask both experts and users to fill up the questionnaires and we compare the results. In the questionnaires, experts are also asked to evaluate how noisy the low level concept are (very reliable, reliable, not reliable).

1.d) Concept is costly to Identify
When a concept is costly, we cannot rely on the fact that the information will be given. For example, if some information on the state of the roots is requested, then the plant must be killed to get the information. Cost could also be associated time elapsed to obtain the information (in air-traffic control, the expert system must take decision before the planes crash into each others) and so on. The cost of performing a test is given by the expert in the questionnaires.

1.e) Uncertainty in the measure of an attribute
When testing numerical parameters, the measuring instrument might not be 100 % accurate. The system should not rely on the outcome of a test like "Is A<B" when A is in the neighborood of B (Zucker. 1978V Clearly, when A gets close to B, we do not want to rely heavily on the outcome of the test. A solution is to . replace the test "A<B" by the two tests "A<B-A" and "A>B+A" where A is given by the expert and ignore the outcome of the test when A is within A of B.

The fundamental idea to treat unreliable data is to delay testing the concept. As a consequence, the noisy tests are performed lower down in the decision tree (and their influence is limited) or not performed at all when there are other alternative tests. Another way to treat this noise is to favor clusters where unreliable information is generalised. *Thus, if in my clusters I have brown spots and brown necrosis (polymorphic low-level concepts), the characteristic description of the cluster will be brown symptom and it will not be noisy anymore.* The result is that the final rules will be more robust in presence of unreliable information.

It is often the case that noisy descriptors are important descriptors that allow discriminating between two similar high level concepts (polymorphy of high-level concepts). Nevertheless it happens thats sometimes the experts wants to show off and use concepts that are very difficult to identify to reach a conclusion while another one would have been simpler and equally good. Note that a common practice when using rules with numeric uncertainty is to lower the certainty factor of rules that contain unreliable premises and the expert does not trust the answer to the questions. Again, this hides the problem as opposed to solving it in a "clean" manner.

V) Randomness of natural phenomena

As surprising as it may sound, we do not consider this as being an important type of noise. Indeed, when learning from examples we assume that the real live case will occur in the training set and do not have to worry about this type of noise.

III.B. Wrong information

Wrong information is human introduced mistakes. We assume that it is not due to unreliable information. This can be:

2.a) Giving the wrong value to a descriptor
For example, the expert might be distracted when he write the description of an example

2.b) Describing a class and attributing the description to another
This can occur for the same reasons as 2.a. However, the effect for the learning system will be much worse.

2.c) Giving too many descriptors
This can haDDen when our evpert uses bv mistake a plant which has several diseases as a reference for a specific disease. It can also happen when the plant presents natural imperfection with are not due to the disease and which are incorrectly identified as a symptom of the disease. Finally, it also happens because people mention concepts that are irrelevant for psychological reasons (Pazzani, 1986).

This can introduce inconsistencies in the knowledge base in which case we can easily detect it and request that the expert solve the problem. Otherwise, errors in the positive examples will cause over-generalizations (Fu, 1985).

A method to handle false positive and negative examples is to allow some uncertainty in the clusters or the rules. This numeric method has been succesfully used in RL (Fu, 1985) or by performing tree-pruning (Quinlan, 1986). In the last one, the error in classification introduced by tree pruning is lower than the error of trees that find total classifications.

2.d) Noise in the background knowledge
We are currently totally empty handed to detect or treat this type of noise. This might be detectable by analysing bad consultations (assuming we have a way of detecting bad consultations, which we do not).

III.C. Incomplete information

This can be:

3a) Forgetting an example
By asking examples to several experts, this problem will be reduced since it is unlikely that all the experts forget the same example. However, as we have seen in section II.B the system checks that the conclusion he obtain match a full description of a concept and learns incrementally if it does not.

3.b) Forgetting a relevant descriptor
This can happen because of lack of attention, the concept is naturally noisy (see IV.A.) or is a default value (this is solved by inheriting properties in our frames). It also happens that sometime people simply do not to mention relevant features for no apparent reasons (Pazzani, 1986).

By using questionnaires, we push the experts to give all the information.

3c) Giving a value for a descriptor which is too general
This can be treated in a similar manner as when the expert forgets the descriptor

3d) High level concept is "fuzzy"
This can be caused by the fact that the outcome of a test which is needed in order to discriminate between some diseases is unknown (lack of information, high cost associated with a test as in 1 .d).

For example, finding which virus causes a disease requires some lab tests. The results of these lab tests will never be known by the person who uses the system. Hence, as far as the system is concerned, it will never be able to discriminate between different viral diseases. Our solution to the problem consists in collapsing the classes into a single one(VIRUS).

3.e) Missing background knowledge
This can be detected and treated when the experts review the characteristic descriptions that have been automatically generated. When they disagree with the generalization, they must explain why and usually this generates new axioms on the domain.

3.f) Missing descriptors in the vocabulary
Consider the following example:

E_1 : [x: <ISA PLANT> <AGE MATURE>] & [spotsl: <ISA SPOT> <NUMBER SEVERAL> <COLOR WHITE> <FORM CONCAVE>] & [fruitl <ISA FRUIT>] & [facel: <PART-OF fruit1> <IS EXPOSED-TO-SUN>] => SUNBURN

E_2: [x: <ISA PLANT> <AGE MATURE>] & [spotsl: <ISA SPOT> <NUMBER SEVERAL> <COLOR YELLOW> <FORM CONCAVE>] & [fruitl <ISA FRUIT>] & [facel: <PART-OF fruit1> <IS EXPOSED-TO-SUN>] => SUNBURN

By climbing the generalization frame, one could conclude that a caracteristic description of SUNBURN is: there are light colored spots on the face of the fruit exposed to the sun.

CE : [x: <ISA PLANT> <AGE MATURE>] & [spotsl: <ISA SPOT> <NUMBER SEVERAL> <COLOR GREY-BEIGE> <EVOLUTE-INTO spots2> <FORM CONCAVE>] & [spots2: <ISA SPOTS> <COLOR BROWN>] & [fruitl <ISA FRUIT>] & [facel: <PART-OF fruitl> <IS EXPOSED-TO-SUN>] & [leavesl: <ISA LEAVES>] & [symptoml: <ISA SYMPTOM>] => NARCOSIS-OF-FRUIT-EXTREMITY

Since grey-beige is also a light color, the negative example is covered. Let us assume that the origin of the problem is the generalization of the color attribute. One can solve this problem by introducing an intermediary level node G0001 in the frame such that WHITE and YELLOW are sons of G0001, G0001 is a son of LIGHT-COLOR and GREY-BEIGE is not a son of G0001. The generalization then becomes COLOR(G0001 ,spots1) which rejects the negative example. This is similar to RL symbolization of taxonomy points (Fu & Buchanan, 1985) and shift of bias (Utgoff, 1986).

This might not be the optimal solution (try to find a meaning for G0001). The correct solution (obtained through interraction with the experts) is that WHITE and YELLOW belong to another frame which had been overlooked, the TRANSLUCENT frame. WHITE and YELLOW are not only colors but they also color loss (when exposed too long to the sun, the colors fade away and become translucent). While in some case one can clearly see that the white or yellow is a color and not a color loss (this means that the proceeding solution does not hold), in some other cases it is more difficult (in other word, we cannot introduce two new colors such as TRANSLUCENT-WHITE and TRANSLUCENT-YELLOW).Of course, multiple links of this sort generate problems for inheriting properties.

In the present case (the SUNBURN disease), the fact there are no symptoms on leaves is also relevant while the fact that the color does not evolve into brown is not. The correct generalization will then be : there are translucent spots on the part of the fruits exposed to the sun and no symptoms on the leaves.

IV. CONCLUSION

When learning from noisy data, it is often though that nothing but numerical coefficients can take into account this uncertainty. We do not claim that one should never use statistical information, but that:

1) The symbolic approach to ML (for example the version space approach) may be usefull in noisy situation when noise comes from a lack of training examples
2) Cross-examination of expertise and comparison of users'descriptions with experts'description can filter noise.
3) finding good intermediary knowledge and knowledge representation, using domain specific knowledge, collecting examples by means of questionnaires can solve some types of noise
4) delaying unreliable tests as much as possible in a decision tree when building a decision tree can generate robust rules.
5) Each time noise is not of numeric nature, as presented in this paper, the introduction of coefficients may lead to some results but that these will not have any significance.

Acknowledgements

We wish to thank Jim Blythe and Rob Corlett at the General Electric Company, Thierry Andro at Cognitech and Jean Jacques Cannat in our group for all the contribution they made to the work described in this paper. Special thanks to Bruce Buchanan and to the Helix group at Stanford University for the useful comments they have made on this research. The learning tools used for this research are implemented in COMMON LISP on a TI-EXPLORER Lisp machine and on a SUN 3/160 workstation.

REFERENCES

(Alexander et al, 1986) Alexander, J. H. & Freiling, M. & Shulman, S. & Rehfuss S. & Messick, S. "Onthological Analysis: An Ongoing Experiment: Proc. of the IstAAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Buchanan & Shortlife, 1984) Buchanan, B. G. & Shortlife, E. H. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addisson-Westley 1984.

(Charniak, 1977) Charniak, E.
"A framed Painting: The Representation of a Common Sense Knowledge Fragment," Cognitive Science, vol 1 #4, pp. 355-394, 1977.

(Clancey, 1986) Clancey, W.
"Transcript of plenary session: Cognition and Expertise." 1st AAA1 Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Corlett, 1983)Corlett, R.
"Explaining induced decision trees," Proc. Expert Systems, pp. 136-142, 1983.

(Fu & Buchanan, 1985) Fu, L M. & Buchanan B. G. Teaming Intermediate Knowledge in Constructing a Hierarchical Knowledge Base", working paper N° KSI 85-4, Computer Science Deparment, Stanford University, 1985.

(Fu, 1985) Fu.L M.
"Learning Object Level and Meta Level Knowledge in Expert Svstems." Phd dissertation, Department of Computer Science, Stanford University 1985.

(Hayes, 1984) Hayes, P.
"Liiquids", in Formal Theories of the Commense World, Hobbs, J. ed Ablex, Hillsdale 1984.
(Kodratoff & Qanascia, 1986) Kodratoff, Y. & Ganascia J. G.
"Improving the Generalization Step in Learning." in Machine Learning 2 : An Artificial Intelligence Approach, Michalski R. S., Carbonell J. G., Mitchell T. M. eds, Morgan Kaufmann, Los Altos 1986.

(Kodratoff et al, 1986) Kodratoff, Y. & Manago, M. & Blythe, J. S. & Smallman, C. & Andro, T.
"Generalization in a Noisv Environment: the Need to Integrate Symbolic and Numeric Techniques in Learning." Proc. of the 1st AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Manago, 1986) Manage M.
"Object-oriented Generalization: A Tool for Improving Knowledge Based Systems," Proc. of the 1st International Meeting on Advances in Learning, Les Arcs (France) 1986.

(Manago & Kodratoff, 1987) Manago, M. & Kodratoff, Y.
"ModelDriven Learning of Disjunctive Concepts." Proc. of the second European Working Session on Symbolic Learning, Bled (Yougoslavia), 1987.

(McDermott, 1986) McDermott, J.
"Transcripts of plenary session; Interactive Tools 1," 1st AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Michalski & Chilauski, 1980) Michalski, R.S. & Chilauski, R.L
"Learning by Being Told and Learning from Examples; An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis." Policy Analysis and Information Systems, Vol 4:2, 1980.

(Minsky, 1975)Minsky, M.
"A framework for representing knowledge," In The Psychology of Computer Vision. Winston P. H. ed, Mc Graw-Hill, New York 1975.

(Mitchell, 1985) Mitchell, T.M.
"Generalization as Search", in Reading in Artificial Intelligence, Webber, B.L, Nilsson, N.J. eds, Morgan-Kaufmann, Los Altos (CA), 1985.

(Nilsson, 1984) Nilsson, N.J.
Principles of Artificial Intelligence, Morgan-Kaufmann, Los Altos (CA), 1984.

(Pazzani, 1986) Pazzani, M.J.
"Explanation-Basad Learning for Knowledge-Based Systems." Proc of the 1 st AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Quinlan, 1983) Quinlan, J. R.
"Learning efficient classification procedures and their application to chess end games." in Machine Learning 1: An Artificial Intelligence Approach, Michalski R. S., Carbonell J. G., Mitchell T. M. eds, Morgan Kaufmann, Los Altos 1983

(Quinlan, 1986) Quinlan, J. R.
Simplifying decision trees." Proc. of the 1st AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Regoczei & Plantinga, 1986) Regoczei, S. & Plantinga, E.O.
"Ontology and Inventory: A Foundation for a Knowledge Acquisition Methodology."Proc of the 1st AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff (Canada) 1986.

(Shank & Carbonell, 1979) Shank, R. & Carbonell, J. G.
"Re: The Gettysburg Address: Representing Social and Political ACtS," in Associative Networks: Representation and Use of Knowledge by Computers. Findler N. V. ed, Academic Press, New York 1979.

(Smith & Medin, 1981) Smith, E. & Medin, D.
Categories and Concepts, Cambridge: Harvard University Press, 1981.

(Utgoff, 1986)Utgoff, P.E.
"Shift of Bias for Inductive Concept Learning." in Machine Learning 2: An Artificial Intelligence Approach, Michalski R. S., Carbonell J. G., Mitchell T. CP eds, Morgan Kaufmann, Los Altos 1986.

(Waltz, 1975) Waltz, D.L
"Understanding Line Drawing of Scenes with Shadows." in The Psychology of Computer Vision, Winston P. H. ed, Mc Graw-Hill, New York 1975.

(Winograd, 1972) Winograd, T.
Understanding Natural Language. Academic Press, New York 1972.

(Zucker, 1978) Zucker, S.
"Production Systems with Feedback," in Pattern Directed Inference Systems, Waterman D., Hayes-Roth F. eds, Academic Press, New York 1978.