88 - 19

Submitted to Machine Learning Journal, revised version (October 20th, 1989)

LEARNING ATTRIBUTIONAL TWO-TIERED DESCRIPTIONS OF FLEXIBLE CONCEPTS

F.Bergadano[1], S.Matwin[2], R.S. Michalski, J. Zhang

Artificial Intelligence Center
George Mason University,
Fairfax, VA 22030
email: jzhang@gmuvax2.gmu.edu

ABSTRACT

Most real-life concepts are *flexible* , that is they lack precise definitions and are context dependent. Representing and learning flexible concepts presents a fundamental challenge for Artificial Intelligence. This paper describes a method for learning such concepts, which is based on a *two-tiered* concept representation. In such a representation the first tier, called the *Base Concept Representation (BCR),* describes the most relevant properties of a concept in an explicit, comprehensible, and efficient form. The second tier, called the *Inferential Concept Interpretation* (ICI), contains procedures for matching instances with concept descriptions, and inference rules defining allowable transformations of the concept under different contexts and exceptional cases.

In the method, the BCR is obtained by first creating a complete and consistent concept description, and then optimizing it according to a general description quality criterion. The complete and consistent description is obtained by applying the AQ inductive learning methodology. The optimization process is done by a double level best first search. The ICI is defined in part by a method of flexible matching and in part by a set of inference rules. The method has been implemented in the AQTT-15 learning system, and experimental results show that such a two-tiered concept representation not only produces simpler concept descriptions, but may also increase their predictive power.

Key words: two-tiered concept representation, inductive learning, simplification of concept descriptions, description quality measure, heuristic search

---

[1] On leave from Universita di Torino, Italy.

[2] On leave from University of Ottawa, Canada.

1. INTRODUCTION

Most current methods of machine learning assume that concepts are precise entities, representable by a single symbolic description. The boundaries of such concepts are well-defined and context-independent. Concept instances are usually assumed to be equally representative. If an instance satisfies the given concept description, then it belongs to the concept, otherwise it does not.

In some methods, these assumptions are somewhat relaxed by assigning to a concept a set membership function (e.g. Zadeh, 1974) or a probability distribution (e.g., Cheeseman et al., 88). However, once such measures are defined explicitly for a given concept, the concept again has a fixed well-defined meaning. Moreover, concept descriptions remain inadequate for handling exceptional cases, for representing context-dependency, or for capturing increases of knowledge about the properties of the concept.

In contrast, most human concepts lack precisely defined boundaries, and have a context-dependent meaning. The imprecision of the boundaries seems to have a logical rather than probabilistic character. It means that the classification of instances of flexible concepts typically involves logical, rather than probabilistic inference. Also, examples of human concepts are usually not all equivalent. They may have different *degrees of typicality* in representing the concept. For example, a robin is conventionally viewed as a more typical bird than a penguin or an ostrich. Also, under different contexts the "bird" concept may apply to a live, flying bird, a picture or a sculpture, a chick hatching out of the egg, or even an airplane. Thus human concepts are *flexible*, as they adapt to the context in which they are used. It is clear that in order to handle such flexible concepts, machine learning systems need to employ richer concept representations than are currently used. Developing methods for acquiring flexible concepts and reasoning with them is thus an important goal in the new phase of machine learning research.

The starting point of the research presented here is the idea of the *two-tiered concept representation* proposed by Michalski (1987). In this representation the total meaning of a concept consists of two components, the *Base Concept Representation* (BCR) and the *Inferential Concept Interpretation* (ICI). The BCR defines the most relevant properties of the concept. The ICI makes the boundaries of the concept flexible by describing allowed modifications of the concept's features in different contexts. Early ideas on learning two-tiered concept representations were presented in (Michalski, 1988). A system which employed a simple form of rule simplification, called TRUNC, to implement two-tiered representation is described in (Michalski et al., 86). An intriguing result of that research was that a substantial reduction of the description's complexity can be achieved without affecting its performance. The effect was obtained through removal of these parts of the description which were responsible for covering only a small fraction of examples (removal of so called *light complexes* from the description), and by applying a flexible matching function for second tier classification.

This paper is an extension and continuation of these early ideas. Important advances are the development of a heuristic search procedure that explores the space of two-tiered descriptions, the use of a more powerful matching procedure, and the introduction of a rule base for performing the ICI. The search is done by applying simplification operators and is guided by a new general description quality measure taking into consideration the accuracy, the comprehensibility and the computational cost of both parts of the two-tiered description (BCR and ICI). By introducing such a general evaluation measure (Bergadano et al., 1988) the learning process can be redefined. Namely, it can be viewed as a multistep process of modifying/improving an initial concept description to maximize the concept quality criterion, which reflects the goals of learning. In such a process the initial description could be, e.g., a set of positive and negative examples, a complete and consistent description obtained by an inductive learning program, or a tentative description supplied by a teacher.

The presented method has been implemented in the AQTT-15 learning system and experimentally applied to two different test problems: learning the concept of an acceptable union contract and learning the distinction between republicans and democrats in the U.S. congress. The experiments have confirmed the initial findings that a two-tiered representation can lead to a substantial reduction of memory and at the same time to an improvement of its predictive power.

2. TWO-TIERED CONCEPT REPRESENTATION

Traditional work on concept representation assumes that the whole meaning of a concept resides in a single stored structure, e.g. a semantic network that captures all relevant properties of the concept (Collins and Quillian, 1972, Minsky, 1975, Sowa, 1984). The process of recognizing a concept involves simple matching of the stored representation with the perceived facts. Such matching may include comparing feature values in instances and concept descriptions, or tracing links in networks of concepts, but has not been assumed to involve any complex inferential processes. A different approach is followed in case-based reasoning and learning systems (Bareiss, 1989), (Hammond, 1989), (Kolodner 1988). In this area of research, domain knowledge, deep indexing and powerful inference mechanisms based on similarities and differences of cases are used in the matching phase, but concept descriptions correspond to individual examples.

The starting point of our approach is the observation that human knowledge can be viewed as a combination of two components, recorded knowledge and inferential extension, i.e., knowledge that can be created from recorded knowledge by conducting many forms of inference. This view leads us to the proposition that the meaning we assign to a concept in any given situation is a result of an interplay between two parts. The first part represents what the agent knows, or remembers. The second part represents what the agent can infer from his knowledge, using rules of inference (deductive, inductive and analogical). Cognitive science supports this point of view in the so called transformational model (Smith and Medin, 1981). In this model, besides classical matching of object features with concept descriptions, features of an object are transformable into features comprising the concept definition. Consequently, the matching process besides being simply a substitution of individual feature values for variables in a concept description, may also have a transformational, or inferential, character.

In order to investigate the consequences of this conjecture, Michalski (1987) has proposed a two-tiered representation of individual concepts. A concept description is split into two-parts: a *Base Concept Representation* (BCR) and an *Inferential Concept Interpretation* (ICI). The BCR defines the concept explicitly, by giving a description of the concept, either in terms of the attributes observed in the example, or in terms constructively learned during concept formation. The prototypical instances of the concept are classified by matching with the BCR. Characteristics of the concept represented in the BCR tend to capture the principle, the most relevant properties or the intention behind the concept.

Anomalies, exceptions and context-dependencies[1] are covered by a reasoning process that uses information contained in the ICI. The ICI deals with exceptions by inferring that they are either extensions of the base representation (concept *extending*), or that they ought to be excluded from the base representation (concept *shrinking*). This process involves the background knowledge and relevant inference methods contained in the ICI, that allow the recognition, extension, or modification of the concept meaning according to context.

When an unknown entity is matched against the BCR, it may satisfy it directly, or it may satisfy

[1]The terms such as anomalous, exceptional, and representative examples, as well as context, are used here in their colloquial meaning. They will be more precisely defined in sec. 2.4.

some of its inferential extensions. During the process of interpreting the ICI, one may use a probabilistic inference based on a simple distance measure (so called *flexible matching* (Michalski et al., 1986)), analogical reasoning, inductive reasoning, or deductive reasoning to classify "special" uses of concepts.

Let us illustrate the idea of two-tiered concept representation with the concept of chair. A two-tiered representation of the chair concept could have the following form:

```
BCR: A piece of furniture.
     Purpose: to seat one person.
     Structure: seat, four legs, and a backrest.

ICI: No-of legs may vary from 1 to 4
     two wheels support seat --> irrelevant(four legs)
     chair without the backrest --> stool
     context = museum exhibit --> chair is not used for seating
               persons
     context = toys --> size can be much smaller. Does not serve
               for seating persons, but correspondingly small
               dolls
```

This simple example illustrates several important features of the two-tiered representation. If recognition time is important, only BCR will be used to match an example. If more time can be allocated, or if a more precise classification is required for a given event, ICI is used. When interpreting the ICI rules, one relies on background and general knowledge, and on the context in which the concept operates. Contexts can have hierarchical organization. Finally, ICI rules may be chained, although it is not shown in this simple example.

2.1. Base Concept Representation

In the method presented, we use the attribute based Logic System VL_1 (Michalski 1983) as a representational formalism. The BCR for a concept is a disjunctive normal form expression in VL_1. Such an expression corresponds to a cover, which is a set of complexes. A complex is a conjunction of selectors. A selector is a form:

[L # R]

where the attribute L is called the *referee*. R, called the *referent.*, is a set of values from the domain of L. Symbol # denotes one of the relational symbols =, <, >, ≤, ≥, ≠.

For example the expression [shape = circle v square] & [length = 2] is a complex containing two selectors. A cover could be obtained by supplying a set of such complexes, which correspond to alternative descriptions. It is obvious how covers correspond to disjunctive normal form expressions.

2.2 Inferential Concept Interpretation: Flexible Matching Function

A flexible matching function F is used as a part of the ICI and it is predefined. The flexible matching function F measures the degree of fit between the event and a concept description. The specific F used in our current implementation matches events from the set E with concept descriptions from the set D:

$$F: E \times D \;\text{-->}\; [0..1].$$

The value of F of an event e and a cover c is defined as the probabilistic sum of F of its complexes. If c consists of a disjunction of two complexes cpx_1 and cpx_2, we have:

$$F(e,c) = F(e,cpx_1) + F(e,cpx_2) - F(e,cpx_1) * F(e,cpx_2)$$

The value of F(e,cpx) in the above expression is substituted by 0 if it is below a given threshold t. The probabilistic sum introduces a strong bias toward the covers that consist of many complexes. If the cover c is represented by many complexes, F(e,c) may be close to 1, even if each F(e,cpx) is very small (see Table 3 in sec. 6).

The degree of fit F of an event e and a complex cpx is defined as the average of the corresponding degrees for its constituent selectors, weighted by the proportion of positive examples covered by the complex:

$$F(e,cpx) = (\textstyle\sum F(e,sel_i)/n) * \#cpxpos/(\#cpxpos + \#cpxneg)$$

where n is the number of the selectors in cpx, and #cpxpos and #cpxneg are the number of positive examples covered by cpx, and the number of the negative examples covered by cpx, respectively.

F of an event e and a selector sel is defined by the degree of match between the selector and the event weighted by the coverage of positive and negative examples of the selector:

$$F(e,sel) = DMatch(e,sel)*(1+(\#selpos/\#pos - \#selneg/\#neg))/2$$

where #selpos and #selneg are the number of positive and negative examples covered by the selector, respectively. #pos and #neg are the number of the positive and negative examples, respectively. Suppose that selector sel is $[x = a_{j_1} \vee \ldots \vee a_{j_m}]$. DMatch(e, sel) is then defined as

$$\mathrm{DMatch}(e, sel) = \begin{cases} 1 & \text{if } x \text{ is nominal and } e \text{ is covered by } sel, \\ 0 & \text{if } x \text{ is nominal and } e \text{ is not covered by } sel, \\ 1 - dis(a_k, sel)/Max & \text{if } x \text{ is linear} \end{cases}$$

where $Max = max_{i=1,\ldots,n}(dis(a_i, sel))$,
$dis(a_k, sel) = min_{i=j1,\ldots,jm}(|i - k|)$

It is assumed that the domain of the selector x is the ordered list $(a_1, a_2, \ldots, a_n)$, and a_k is the value of x of the event e. For example, if the domain of x is [0 ... 10] and the value of x for the event e is 4, then
$DMatch(e,[x = 2v5]) = 1- (5-4)/max_{i=0,\ldots,10}(dis(i, sel)) = 1 - 1/5= 0.8$.

The system is not forced to make a decision when the difference between the values of flexible matching function for two concepts is very small. If the difference is smaller than the preset threshold, the result will be no_match.

2.3 Inferential Concept Interpretation: Deductive Rules

In addition to flexible matching, the ICI includes a set of deductive rules, allowing the system to recognize exceptions and context-dependent cases. For example, flexible matching could allow us to recognize a sequoia as a tree, although it does not match the typical size requirements. Deductive reasoning is required to recognize a tree without leaves (in the winter time) or to include in the concept of tree its exceptional instance (e.g. a fallen tree). In fact, flexible matching is most useful to cover instances that are close to the typical case, while deductive matching is appropriate to deal with concept transformations necessary to include exceptions or context-dependencies in the concept description.

The deductive rules in the ICI are expressed as Horn clauses. Inference on these rules is implemented using the LOGLISP inference system (Robinson and Sibert, 1982). Numerical quantifiers and internal connectives are also allowed (Michalski 1983).

2.4. Types of Matching

According to the type of knowledge exemplified by it, an event can be covered by a two-tiered description through the following three types of matching:

1. *Strict matching*: the event matches the BCR exactly, in which case we say that the event is S-covered,
2. *Flexible matching*: the event matches the BCR through a flexible matching function, and we say that the event is F-covered.
3. *Deductive matching*: the event matches the concept through deductive reasoning by using the ICI Rules, and we say that the event is D-covered.

These three sets are made mutually exclusive: if an event is S-covered, then it is not D-covered or F-covered, and if an event is D-covered, then it is not F-covered. Thus, S-covered events are explicitly covered, and F-covered and D-covered events are implicitly covered. The sets of all S-covered, D-covered, and F-covered events will be called Scov, Dcov, and Fcov, respectively. We can now give a precise meaning to the term *exception* used throughout this paper. An event is an exception if it neither S-covered nor F-covered. Events that are S-covered are said to be *representative* examples. The notion of representative examples and exceptions depends on the context in which the examples are obtained. Context can be therefore viewed as an environment providing examples together with their typicality.

Two-tiered concept descriptions are usually simpler, easier to understand and more efficient to use than the conventional ones. They also exhibit performance improvement on a testing set. In the systems developed so far, the ICI includes only a flexible matching function. More importantly, in their quality evaluation measures, these early systems do not take into account the inferentially covered parts of concept descriptions. Improvement in quality is therefore measured only by the improvement in the first tier.

3. SYSTEM OVERVIEW

We have implemented a system that produces a two-tiered description of flexible concepts. The system operates in two phases, described below. Table 1 specifies the input, output, and the function of the system.

Learning two-tiered concept descriptions is performed in two stages; the general structure of the system is given in Fig. 1. In our approach, we have relied on AQ15 (Michalski et al. 86) to obtain a complete and consistent concept description. The description generated in this phase, together with the flexible matching function, forms the initial BCR. The second phase improves this initial description by conducting a "double level" best first search. This process is guided, at the first level, by the description quality measure, defining which descriptions should be considered first, and, at the second level, by heuristics that indicate which search operators should be applied to the chosen description.

Phase 1

Given:

Examples obtained from a source.

Background knowledge

Determine

Complete and consistent description of the concept

Phase 2

Given

Complete and consistent description of the concept
Measure of description quality
Background Knowledge

Determine

Two-tiered description, maximizing the description quality measure

---

Table 1. A specification of the system

One can observe that the quality does not have to be measured using the same set of examples from which the initial description was learned during the first phase. A different set of examples could have been used, if the initial description was obtained independently. According to the nature of the quality measure, descriptions can be improved mainly by increasing their accuracy or by decreasing their complexity. For this reason the operators in the search simplify the BCR of a given description by removing some of its components or by modifying the arguments of some predicates. The search process is described as follows:

Search space:	a tree structure, in which the nodes are two-tiered descriptions (BCR + ICI) of a given concept.
Operators:	selector removal, complex removal, referent modification.
Search strategy:	controlled by the quality measure.

The goal of this procedure is not necessarily to find an optimal solution, i.e. the description with the highest quality, because this would require a combinatorial search. On the contrary, the system tries to improve the given concept description by expanding a limited number of nodes in the search tree, and is guided by heuristic information. In the current implementation these heuristics are based on the coverage of the individual selectors and complexes.

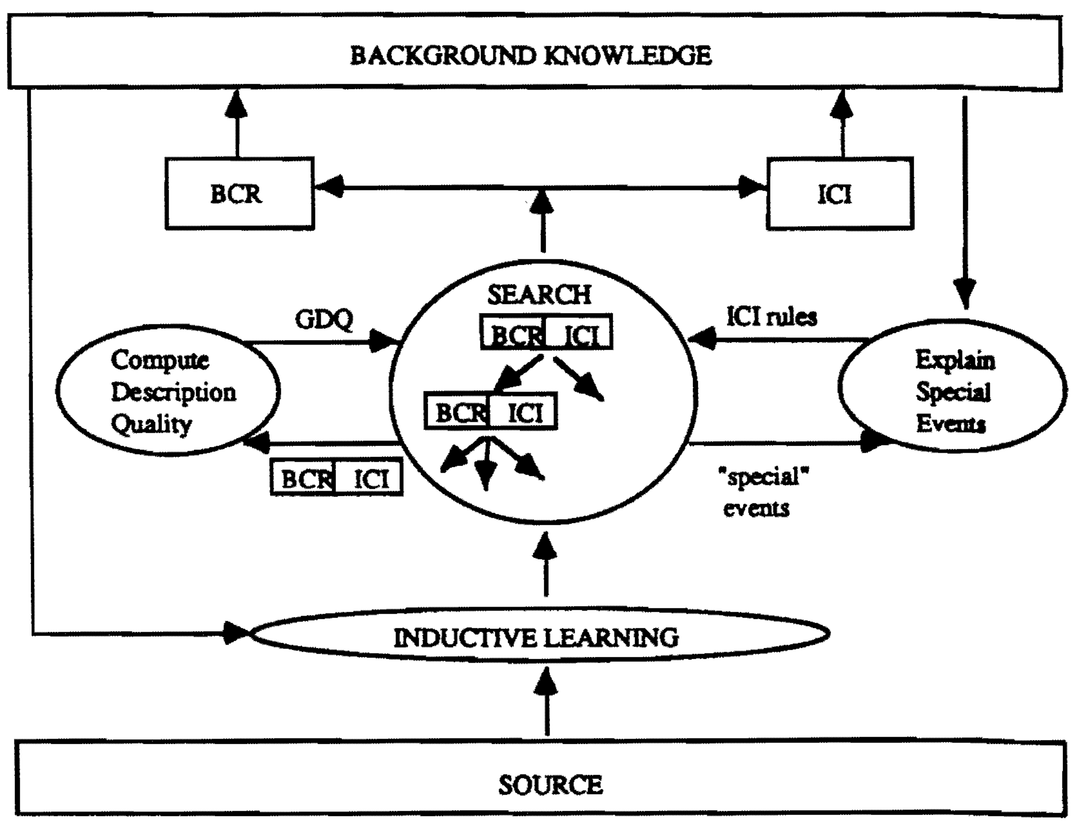

Fig. 1. Design of the AQTT-15 System

The BCR is represented in disjunctive normal form and described in VL_1 notation (Michalski 83). The ICI consists of two parts: a flexible matching function and deductive rules. The system learns the BCR and guides the generation of the deductive rules. The flexible matching function is the one given in the previous section and is fixed during learning. The experimental system relies on the expert to provide either rules that explain special events or a more general domain theory.

3.1. Learning the Base Concept Representation.

An elementary search operator may either specialize or generalize a description. The heuristics used at a given step of the search (section 5.2) decide which operation is applied. In the described system, generalization is done by removing selectors, and specialization is done by removing complexes. Removing a selector is an instance of the "dropping condition" generalization rule (Michalski, 1983). Removing a whole complex is the reverse of the "adding an alternative" generalization rule, and thus is a specialization rule. When a description is specialized, it will then cover a smaller number of positive and negative examples. Another operation, referent modification, simplifies the range of a selector, and may behave either as a generalization, or a specialization, depending on the selector relation (see table II). For example, if the selector is

```
[size = 1..5 v 7]
```

then referent modification giving

```
[size = 1..7]
```

is a generalization, since the cover is extended. On the other hand, if the selector is

```
[size <> 1..5 v 7]
```

then the same referent modification represents a specialization, since the cover shrinks. Table II summarizes the implementation of generalization and specialization operators in the existing system.

Search operator	Knowledge modification performed
Selector removal (SR)	Generalization
Complex removal (CR)	Specialization
Referent extension (RE)	Generalization
Referent shrinking (RS)	Specialization

Table II. Implementation of the search operators

3.2. Learning the Inferential Concept Interpretation.

Each application of a search operator (SR, CR, RE, RS) modifies the BCR, making it either more general or more specific (as described in the previous section). If the new BCR is more specific, some positive events previously covered may not be covered any more. Their coverage can be achieved by the ICI. On the other hand, if the new BCR is more general than the original one, some new events, previously not covered by the BCR, may have been added. These events could be positive as well as negative. If additional negative events are covered as a result of some generalization, they will have to be excluded from the set of events covered by the BCR by means of the ICI rules. Therefore, there are two types of rules in the ICI: rules that cover a positive example otherwise left out of the BCR, and rules that eliminate a negative example from the BCR.

In order to exclude or cover an event by the ICI part of a concept description, one has to obtain rules that will match the event, and perform the action necessary for the exclusion or coverage of the event. These rules, or their chains that ultimately lead to a conclusion regarding the membership of an event in the concept, are treated as an explanation of the event.

Rules can be either inherited from higher level concepts or supplied by the expert. But even if the expert provides a knowledge base which can be used to cover or exclude some examples, the form of this knowledge may not be operationally effective, and may be made more efficient through a process of analytic learning (e.g. Mitchell et al., 86; Prieditis and Mostow 1987). If the knowledge supplied by the expert is too specific or even partially incorrect, it may be improved by induction (Dietterich and Flann 1988, , Mooney and Ourston 1989). In our approach, only exceptions will be subject to explanation. The purpose of the explanation here is to justify the special character of the event explained, rather than to operationalize the proof of its membership in the concept. In both cases the search procedure described in Fig. 1 will guide learning by pointing at the examples that need special explanation.

4. QUALITY OF CONCEPT DESCRIPTIONS

The learning methodology described here is based on the General Description Quality measure (GDQ), that guides the search for better two-tiered descriptions. The quality of a concept description used in our system is influenced by three basic characteristics: the *accuracy*, the *comprehensibility*, and the *cost*. This section discusses these three components, as well as a mechanism for combining them into a single measure.

The accuracy represents the description's ability to produce correct classifications. A common way to prefer more accurate descriptions is to require that they be complete and consistent with respect to the training events (Michalski, 1983). Even if a description is incomplete and inconsistent, the number of positive and negative examples it covers provides important information for evaluating its quality. In this case, we can measure the degree of completeness and consistency of a given description. If the description is also sufficiently general and does not depend on the particular characteristics of the training events, these measures can be a meaningful estimate of the accuracy of the description. In order to achieve completeness and consistency in the presence of noise, one may generate overly complex and detailed descriptions. Such descriptions, however, tend to be too dependent on the training set and consequently may not perform well in future cases and examples. This phenomenon is well known in statistics as overfitting (Watanabe, 1969; Sturt, 1981).

The comprehensibility of the acquired knowledge is related to subjective and domain-dependent criteria. Because an AI system is often supposed to supply advice to humans, knowledge used by such a system should be understandable by human experts. A black box classifier will not be accepted by experts as a help in their work. Therefore, knowledge acquired by a learning system should be related to terms, relations and concepts used by experts, and should not be syntactically too complex. This is called the *comprehensibility* criterion (Michalski, 1983). There is no, however, established measure of comprehensibility of a description. In our method, we will approximate it by representational complexity of the description's expression. This complexity is evaluated by counting the number of operators involved, and taking into consideration the complexity of the operators. Comprehensibility of a two-tiered representation takes into account the operators occurring in both BCR and ICI, and has to weigh the relative contribution of each part to the comprehensibility of the whole description.

The cost captures the properties of a description related to its storage and use (computational complexity). Other things being equal, descriptions which are easier to store and easier to use for recognizing new examples are preferred. When considering the cost of a description, two characteristics are of primary importance. The first one is the cost of measuring the values of variables occurring in the description. In some application domains, e.g., in medicine, this may be a very important consideration. The second one is the computational cost of evaluating the description. Again, certain applications in real-time environment, e.g., speech or image recognition, may impose constraints on the evaluation time of a description. The cost (approximated by computational simplicity) and the comprehensibility (approximated by representational simplicity) are usually related to each other, but in general these are different criteria.

The criteria discussed above can also be applied to two-tiered descriptions. The accuracy of the acquired knowledge does not only depend on the explicit information, but also on the implicit reasoning abilities. Inferential Concept Interpretation also affects cost, since it allows the performance system to use a simpler BCR, and reason about special details only in exceptional cases. Finally, the comprehensibility of a two-tiered representation must be carefully evaluated, since one of its implied goals is to state a clear and simple concept description in the BCR and to account for special cases through a reasoning process. In fact, both the BCR and the ICI are parts of the concept description and they are used together in concept recognition. They influence each other. It is not necessarily true that if a BCR performs well with one ICI, then it also performs well with a different ICI. For example, the experiments described later showed that a BCR performed poorly with an empty ICI, but it performed well with a flexible matching function. Furthermore, in order to learn a better two-tiered concept description, the distribution between the BCR and the ICI should be adjusted during learning. Therefore they should not be learned separately and they should be related during learning. Current existing learning systems acquire a base concept representation from given examples and then may use a form of flexible matching only in the performance element. In our approach, when computing GDQ of a concept description, both the BCR and the ICI are considered. In particular, candidate concept descriptions are compared on the basis of how well they perform with flexible matching. This is only reasonable since flexible matching is actually used during classification of further examples.

These criteria need to be combined into a single evaluation procedure that can be used to compare different concept descriptions. A possible solution is to have an algebraic formula that, given the numeric evaluations of individual criteria, produces a number that represents their combined value. Examples of such approaches are multiplication, weighted sum, maximum/minimum, t-norm/t-conorm (Weber, 1983). Although these approaches are often appropriate, they have certain disadvantages. First, they usually combine a set of heterogeneous evaluations into a single number, and the meaning of this final number is hard to understand for a human expert. Second, they may force the system to evaluate all the criteria, even if it would be sufficient to compare two given descriptions on the basis of the most important one, if one is much better than the other.

In order to overcome some of these problems, we use a lexicographic evaluation functional (LEF) (Michalski, 83) that combines the above mentioned criteria.

4.1. The Preference-based Evaluation Criterion

In this section, we are describing a LEF/WEF method that combines the simple description evaluation criteria. Let us discuss *lexicographic evaluation functional* (LEF) first. The General Description Quality measure under LEF is thus defined as:

$$\text{GDQ(description)} = \langle(\text{Accuracy},\tau_1), (\text{Comprehensibility},\tau_2), (\text{Cost},\tau_3)\rangle$$

where τ_1, τ_2, and τ_3 are tolerance thresholds.

In this evaluation scheme, the criteria are ordered according to their importance, and a tolerance threshold is associated with each criterion. If the difference of the evaluation of two expressions under a given criterion is less than the corresponding tolerance, the two descriptions are considered equivalent with respect to that criterion. The most important measure in the LEF is evaluated first, and the subsequent measure is evaluated only if the previous one is a tie.

The LEF evaluation scheme is not affected by the main problems of algebraic functions which we have discussed above. The importance of a criterion depends not only on the order in which it is evaluated in LEF evaluation scheme, but also on its tolerance. It may be difficult to determine this tolerance. If the tolerance is too small, we have little chance of using the other criteria. If the tolerance is too large, some important criterion might be underestimated. Furthermore, in the case of a large tolerance, many descriptions might be equivalent under the LEF evaluation scheme. In order to avoid this problem, the LEF measure can be extended in the following way: LEF is first applied with larger tolerances, in such a way that all the relevant criteria are taken into account. If the comparison still results in a tie, a Weighed Evaluation Functional (WEF) is used to combine the measures (i.e. the description having the maximum weighted sum of the measures is preferred). The weights for WEF are determined by the user.

4.2. The Role of the Typicality of the Examples.

The accuracy is the first criterion of concept description quality. Accuracy depends linearly on completeness and consistency of the description, as well as on the typicality of the events covered by the two parts of the description. In evaluating the accuracy of a two-tiered concept description, we have to take into account the fact that degree of confidence in the results of inference decreases from deduction to induction (Michalski, 1987). These requirements are met by introducing the notion of typicality (Rosch and Mervis, 1975). Completeness and consistency are made dependent on the typicality of the covered examples and on the way these examples are covered. We assume that an expert can provide typicality of examples at the time they are presented to the system responsible for building the initial description. The experts are usually quite good at determining the typicality of events in their area of expertise.

In general, descriptions that cover many typical positive events are most preferred. Completeness is therefore proportional to the typicality of the events covered. Moreover, if negative events are covered, the consistency of the description is smaller if the typicality of the negative events covered is high.

Completeness and consistency of a two-tiered description brings up additional requirements: a good description should cover the typical examples explicitly, and the non-typical ones implicitly. It is also preferred that the typical events are covered by the BCR, and non-typical, or exceptional events are covered by the ICI. In fact, the BCR is inductively learned from the events provided by user, and it is more reliable when the training events are typical. The ICI, on the contrary, is deductively obtained from the background knowledge, or from a human expert, and relies more on general and domain knowledge. Generally, the ICI is more reliable when dealing with the special or rare cases, since experts often have difficulty in explaining large quantities of typical events. For these reasons, a typical positive explicitly-covered event should contribute to completeness more than implicitly-covered. And vice-verse, non-typical positive implicitly-covered events contribute to completeness more than explicitly-covered. These assumptions are reflected by weights w_s, w_f, w_d, used in the definitions of completeness and consistency (sec. 4.3).

Furthermore, since ICI rules are obtained from background knowledge or from a human expert, they are more reliable than the flexible matching function. Consequently, a positive D-covered event should contribute to completeness more than F-covered. We may also observe that flexible matching is not very useful for exceptions of low typicality. A similar argument holds for consistency.

4.3. A Detailed General Description Quality Measure.

The purpose of this section is to define in detail the GDQ measure that was implemented in our experimental system. First, we have to define the completeness with typicality measure (COMPT) and the consistency with typicality measure (CONST) of a description:

$$\text{COMPT} = \frac{\sum_{e+\in S\text{-cov}} w_s * \text{Typicality}(e^+) + \sum_{e+\in F\text{-cov}} w_f * \text{Typicality}(e^+) + \sum_{e+\in D\text{-cov}} w_d * \text{Typicality}(e^+)}{\sum_{e \in POS} \text{Typicality}(e)}$$

Typicality-dependent consistency (CONST) of a description is defined as follow:

$$\text{CONST} = 1 - \frac{\sum_{e-\in S\text{-cov}} w_s * \text{Typicality}(e^-) + \sum_{e-\in F\text{-cov}} w_f * \text{Typicality}(e^-) + \sum_{e-\in D\text{-cov}} w_d * \text{Typicality}(e^-)}{\sum_{e \in NEG} \text{Typicality}(e)}$$

where POS is the set of positive events covered by a two-tiered concept description, NEG is the set of negative events covered by a two-tiered concept description, and Typicality(e) is the numeric degree of typicality of the event e specified by the expert when the event is given. Furthermore, weights assigned to different types of coverage depend on thresholds to reflect the appropriateness of types of coverage for different kinds of typicality:

w_s: if $\text{Typicality}(e) \geq t_2$ then 1, else w,

w_f: if $t_2 \geq \text{Typicality}(e) \geq t_1$ then 1, else w,

w_d: if $t_2 \geq \text{Typicality}(e)$ then 1, else w,

where t_1 and t_2 are thresholds, and $1 \geq t2 \geq t1 \geq 0$, $1 \geq w > 0$.

Now *accuracy* can be defined in terms of COMPT and CONST:

Accuracy(description) = w_1 *COMPT(description) +w_2 *CONST(description)

where $w_1 + w_2 = 1$. The weights w_1 and w_2 reflect the expert's judgement about the relative importance of completeness and consistency for the given problem.

A measure of comprehensibility of a concept description is difficult to define. We will approximate it by a complexity measure, defined as:

$$v_1 * \sum_{op \in BCR(dsp)} C(op) \quad + \quad v_2 * \sum_{op \in ICI(dsp)} C(op)$$

where BCR(dsp) is the set of all operator occurrences in the BCR, ICI(dsp) is the set of all operator occurrences in the ICI, and C(op) is the complexity of an operator. Complexity of an operator is a real function that maps each operator symbol into a real number. Values of complexity of operators are ordered as follows:

C(interval) < C(internal disjunction) < C(=) < C(<>) < C(&) <

C(v) < C(implication).

When the operator is a predicate, C increases with the number of the arguments of the predicate.

In the above expression, v_1 and v_2 are weights, and $v_1+v_2 = 1$. The BCR should describe the general and easy-to-define meaning of the concept, while the ICI is mainly used to handle rare or exceptional events. As a consequence, the BCR should be easier to comprehend than the ICI (v_1 should therefore be larger than v_2).

The cost consists of two parts:

Measure-Cost -- the cost of measuring the values of variables used in the concept description, it is defined as the function MC

Evaluation-Cost-- the computational cost of evaluating the concept description, it is defined as the function EC.

$$MC(description) = \sum_{e \in Pos \cup Neg} \sum_{v \in vars(e)} mc(v)/(|Pos| + |Neg|)$$

$$EC(description) = \sum_{e \in Pos \cup Neg} ec(e)/(|Pos| + |Neg|)$$

where vars(e) is the set of all occurrence variables used to evaluate the concept description to classify the event e, mc(v) is the cost of measuring the values of the variable v, and ec(e) is the computational cost of evaluating concept description to classify the event e. The latter depends on computing time or on the number of operators involved in the evaluation.

We now define the cost of a description:

$$Cost(description) = u_1 * MC(description) + u_2 * EC(description)$$

where u_1 and u_2 are weights defining the relative importance of measure cost and evaluation cost.

With the exception of the weights which can be determined experimentally, we have already defined all three components of the quality measure of concept descriptions. More details and example about the quality measure can be found in (Bergadano et al. 1988).

5. LEARNING BY MAXIMIZING THE CONCEPT DESCRIPTION QUALITY

Learning two-tiered concept descriptions is performed in two phases. In the first stage, a complete and consistent concept description is obtained from an inductive learning system. In our approach, we have relied on AQ15 (Michalski et al., 86a) to obtain such descriptions. This paper concentrates therefore on the second phase, i.e. the improvement of the description obtained in the first phase with respect to its GDQ, through a process of simplification.

5.1. Search Heuristics

The GDQ measure is computationally expensive, because it requires the system to perform flexible matching for every newly generated description against the whole set of training examples. In order to limit this inefficiency we have introduced a heuristic measure for choosing the operation which has the best chance of improving the GDQ of the description. The specific choice of the operators CR and SR is determined based on the *Potential Accuracy Improvement* heuristic (PAI). This scheme corresponds to a double-level best first search: the GDQ is used to select the description to be modified by the search operators (the node to be expanded) and the PAI is used to select one of the operators, which will be then applied to the selected description. The idea behind the PAI heuristic is to truncate first a complex which covers uniquely a small number of examples. Then, a specific choice of the operator S_{ij} is made so that the removal of a selector improves the completeness of the description, while the consistency measure has an acceptable value. Finally, when no other operator is recommended, referent modification can be selected. Referent modification can improve both consistency and completeness measures. In the worst case, this algorithm will perform a PAI-controlled exhaustive search of the whole search space.

The complex and selector truncation heuristics are implemented together in the following way. Let us first define the PAI. The PAI of truncating a complex is defined as follows:

PAI = #CNI/#NEG - #CPI/#POS

where #CNI (#CPI) is the number of negative (positive) examples no longer covered by the concept description after truncating the complex, respectively. #NEG and #POS are the numbers of negative and positive examples, respectively. The PAI of truncating a selector is more complex and is defined as follows:

PAI = (#SPI/#POS) * P - (#SNI/#NEG) * N

where #SNI (#SPI) is the number of additional negative (positive) examples covered by the concept after having truncated the selector, respectively. N is the proportion of the negative examples which are not covered by the description, and P is the proportion the positive examples which are not covered by the description.

The PAI heuristic is more efficiently computed than the GDQ. For every selector in the descriptions we maintain the list of examples covered by it, using bit vectors. The list of examples covered by complexes and covers will then be obtained from this by intersection or union operations operations. Matching time can be improved further by maintaining also the bit vectors for the examples covered by complexes (time for intersection will be traded off against memory). Note that the GDQ could not be obtained by intersection and union of bit vectors, since it requires flexible matching.

If an even more efficient heuristic is needed, the following is used:

`PAI' = #NI/#NEG - #PI/#POS`

where `#NI (#PI)` is the number of negative (positive) examples covered by the complex or the selector to be removed. This heuristic information is efficiently computed because it can be obtained before the search starts, for every selector and every complex in the initial description, and does not need to be repeated for every node in the search, as is the case for the GDQ measure.

The operator is chosen based on the value of `PAI`. The operator with the largest `PAI` is chosen. Finally, the `PAI`s of selector and complex removal are weighted differently. More weight is assigned to `PAI` of complex removal, since the operator simplifies the description more than selector removal. Nevertheless, it should be noted that removing a selector may enable the system to remove additional complexes afterwards. In fact, by removing a selector from a complex, more examples will be covered, and other complexes may become redundant, because they only cover examples that are already covered. As a special case, two complexes may become syntactically equal after selector truncation, and may be merged into a single one. If two complexes, after truncation, become very similar to one another, they may be merged into a single one by adding alternatives in the internal disjunctions of their selectors. For example the complexes `[shape = circle]&[size = 2]` and `[shape = square]&[size = 2]` may be replaced by `[shape = circle v square]&[size = 2]`.

5.2. Algorithm

The general search procedure is more precisely presented by the following search algorithm:

1. Identify in the search tree the best description D (one with the highest GDQ). Initially, D is the complete and consistent description obtained in stage 1.

2. Apply to D the operator (from among CR_i, SR_{ij}, RE_{ij}, RS_{ij}) that potentially improves GDQ of D the best, based on the Potential Accuracy Improvement (PAI) heuristics

    CR_i: Remove the i-th complex from D.

SR_{ij}: Remove the j-th selector from the i-th complex in D .
RE_{ij}: Extend the referent of the j-th selector in the i-th complex in D.
RS_{ij}: Shrink the referent of the j-th selector in the i-th complex in D.

3. Compute the GDQ of the node obtained in step 2. If this GDQ is smaller than the GDQ of D, then proceed to step 1. As it was mentioned above, when computing the accuracy of a description, flexible matching is always used.

4. Ask for an explanation of
   (a) the positive examples that are not covered any more
   (b) the negative examples that are now covered

   If such an explanation is given, add the rules that make up this explanation to the ICI.

5. Update the GDQ value of the new node, by taking into account the added ICI rules.

6. If the *stopping criterion* is satisfied, then STOP, otherwise proceed to step 1.

We shall now discuss the motivation and details of the algorithm, and explain the search strategy.

In step 1, the nodes are chosen on a best first basis, that is the node in the search space with the highest GDQ value is expanded first. This is not always an optimal choice, since apparently "bad" nodes can lead to better descriptions after a number of removals. Whether the search will behave in this manner will depend on the adequacy of the GDQ as the measure of concept quality.

In step 2, a search operator is chosen heuristically and applied to the description. The heuristics used are discussed in the next section. Only one operator is applied at any given time.

In step 3, the system computes the GDQ of the new node. It should be noted that, in the GDQ measure, the typical examples covered directly by the BCR can weigh more than those covered through flexible matching. The examples covered by ICI rules should weigh more than the ones covered through flexible matching but less than the ones covered by the BCR.

In step 4, the "explainer" module is used in order to improve the description even further: the BCR description is extended or shrunk by adding ICI rules. Firstly, complex removal might have caused some positive examples, that were previously covered, to be lost. In this case some new rules could be introduced in the ICI, that would allow the system to reason about such "special" positive examples, and understand why they should still be classified as instances of the concept under consideration. On the other hand, selector removal might have caused some negative examples to be covered, and new rules in the ICI may be added in order to "shrink" the BCR and avoid these erroneous classifications. Another issue, concerning step 4, is whether an explanation should be required at all, since, in some cases, the chosen removal operator is not an appropriate one, and will lead to a very poor description. In this case it is not even worth to ask for an explanation, and search can continue in other directions. The current strategy is as follows. Suppose the relation < denotes the GDQ ordering among two-tiered descriptions, n is the node we are expanding and m is the node we obtain after the selected removal. If m<<n, then no explanation is even tried, otherwise the explainer is asked for an explanation and is told how m compares to n with respect to <, in order to know how important the request for the explanation is for the search procedure.

In step 5, the GDQ of the obtained two-tiered description is updated after the new ICI rules have been added. Since ICI rules are taken into consideration in the GDQ, new ICI rules will change the GDQ value for a concept.

In step 6, the system decides whether to stop or continue the search. The *stopping criterion* is satisfied when the search space that has been explored is large (more than k1 nodes have been expanded), or when no qualitative improvement has been obtained for a long time (more than k2 nodes have been expanded since the last GDQ improvement). When the system stops, the best node in the search space is produced and becomes the modified two-tiered concept description.

One more characteristic of the system should be mentioned: only one operator is applied to the selected (best-quality) node at any one time; therefore, the new node can be selected if its quality is better than the quality of the father node. This is different from standard search procedures, where all the applicable operators are used for the selected node (node expansion). This choice was introduced because the creation of a new node involves the computation of its quality, which, in some cases, can be time-consuming. On the contrary, we try to avoid generating bad quality nodes by selecting the best applicable operator on a heuristic basis, and applying only that operator. The other operators will be used only if the results obtained on this search branch turn out to be unsatisfactory.

5.3. A Simple Example

An abstract example of the search process is given in Fig. 2. The nodes contain BCR, ICI, and a graphical representation of the covered examples. The tree is kept in memory throughout the search. The BCR is expressed in disjunctive normal form (it is a "cover").

In the example, the accuracy is computed according to the formula discussed in Section 4, assuming the same typicality for all the instances. The initial description is represented in node 1, and contains two disjuncts (complexes). The complexes cover the two corresponding rectangular areas in the graphical representation, containing five positive examples out of eight, and one negative example out of five. The ICI extends this coverage by recognizing one more positive example. By eliminating conjunct (selector) s5 in the second complex we obtain node 3 in the search tree. The accuracy of the description is now improved since all the positive examples are covered. Finally, by truncating the first complex we obtain node 5. It does not cover negative examples any more, and is definitely simpler. This node is then accepted as the improved description resulting from the search. The other nodes lead to inferior concept representations, with respect to GDQ, and are discarded. The quality has been computed with $w_1=w_2=0.5$.

For clarity, the cost is omitted, and the simplicity of the ICI is not taken into account. The simplicity of the BCR depends on the number of complexes and selectors.

6. EXPERIMENTS

We have run experiments with the system in two different domains: labor management contracts and congressional voting record. In particular, the system acquired discriminant descriptions for

(a) acceptable/unacceptable labor management contracts and

(b) republican/democrat congresspersons in the U.S. House of Representatives.

The labor-management contract data used in this section comes from *Collective Bargaining* - a review of current collective bargaining issues published by the Government of Canada through its Department of Labor. The data given in *Collective Bargaining* describes labor-management contracts which have been currently negotiated between organizations and those union locals that count at least 500 members. The raw data is divided geographically, as well as by economic sectors. The format of the raw data is pretty standard. Each contract is described by a number of attributes. Since the attributes vary between economic sectors, we have decided to focus on a single sector: personal and business services. This sector includes unions representing hospital staff, teachers, university professors, social workers, and certain classes of administrative personnel of different organizations. With this kind of data, described by attributes only, it was natural to represent concepts using the VL_1 formalism.

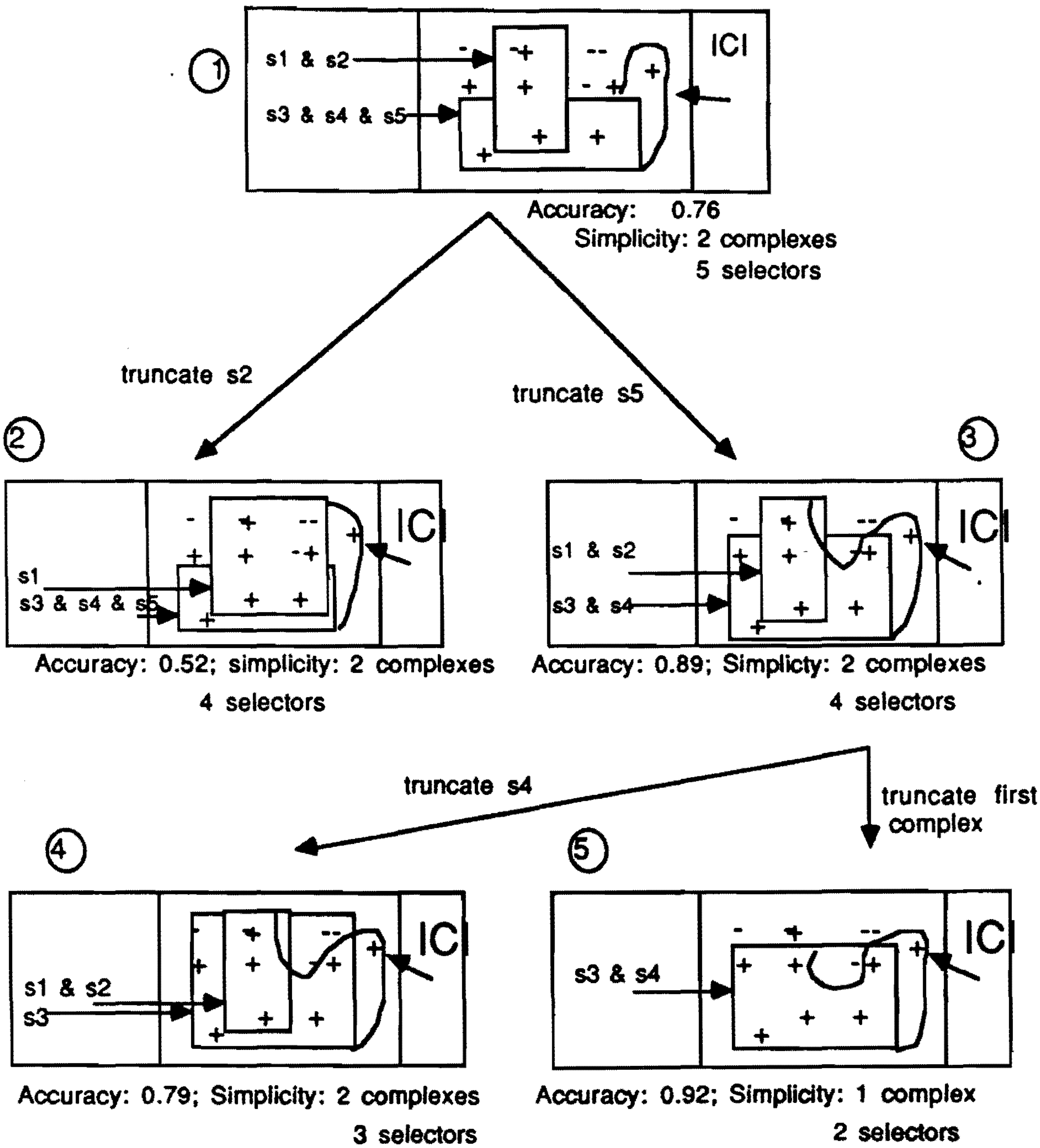

Fig. 2. Example of the search space organized as a tree.

Our data describes contracts finalized in the second half of 1987 and first half of 1988. Each contract is described by sixteen attributes, belonging to two main groups: issues related to salaries (e.g. pay increases in each year of contract, cost of living allowance, stand-by pay, etc.), and issues related to fringe benefits (e.g. different kinds of pension contributions, holidays, vacation, dental insurance, etc.). Negative examples were obtained from description of a contract proposals deemed unacceptable by one of the parties. The training set consisted of 18 positive and 9 negative examples of contracts; the testing set consisted of 19 positive and 11 negative examples.

The second application was concerned with the U.S. Congress voting record. We have relied on the same data set as used by (Lebowitz 1987) in the experiments on conceptual clustering. The data represents the 1981 voting record for 100 selected representatives. The data set was split randomly into a training and testing set, with voting records of democrats entered as positive examples, and voting records of republicans entered as the negative ones. The goal was to obtain discriminating descriptions of democrat and republican congressmen.

Four experiments have been performed. In each experiment we have looked at the number of events correctly and incorrectly covered by both descriptions, and at the number of events that were not covered by either concept. This was done both on the training set and on a testing set of examples not previously seen by the system. The results of the three experiments summarized in Tables 3, 4, 5, and 6 are discussed below. In each of the tables, *L* denotes the labor-management experiment, and *C* denotes the congress experiment. In each experiment, the same training and testing sets were used.

Factual Knowledge
Labor-mgmt data (*L*) : 27 complexes and 432 selectors
Congress data (*C*) : 51 complexes and 969 selectors

		Correct		Incorrect		No_Match	
Strict Match		*L*	*C*	*L*	*C*	*L*	*C*
	Training Set	100%	100%	0%	0%	0%	0%
	Testing Set	0%	4%	0%	0%	100%	96%
Flexible Match							
	Training Set	100%	100%	0%	0%	0%	0%
	Testing Set	37%	66%	0%	2%	63%	32%
1-nearest Neighbor							
	Training Set	100%	100	0%	0%	0%	0%
	Testing Set	77%	85%	23%	15%	0%	0%

Table 3 Results of Experiment 1

Initial Description
Labor-mgmt data (*L*) : 11 complexes and 28 selectors
Congress data (*C*) : 10 complexes and 32 selectors

		Correct		Incorrect		No_Match	
Strict Match		*L*	*C*	*L*	*C*	*L*	*C*
	Training Set	100%	100%	0%	0%	0%	0%
	Testing Set	80%	86%	17%	14%	3%	0%
Flexible Match		*L*	*C*	*L*	*C*	*L*	*C*
	Training Set	100%	100%	0%	0%	0%	0%
	Testing Set	80%	86%	17%	14%	3%	0%

Table 4 Results of Experiment 2.

Base Complex (rule simplification using the TRUNC method)
Labor-mgmt data (*L*) : 2 complexes and 6 selectors
Congress data (*C*) : 2 complexes and 6 selectors

		Correct		Incorrect		No_Match	
Strict Match		*L*	*C*	*L*	*C*	*L*	*C*
	Training Set	52%	62%	0%	0%	48%	38%
	Testing Set	63%	69%	7%	7%	30%	24%
Flexible Match							
	Training Set	81%	75%	19%	25%	0%	0%
	Testing Set	83%	85%	17%	15%	0%	0%

Table 5 Results of Experiment 3.

Optimized Description
Labor-mgmt data (*L*) : 9 complexes and 12 selectors
Congress data (*C*) : 10 complexes and 21 selectors

		Correct		Incorrect		No_Match	
Strict Match		*L*	*C*	*L*	*C*	*L*	*C*
	Training Set	63%	84%	0%	0%	37%	16%
	Testing Set	43%	73%	3%	4%	54%	23%
Flexible Match							
	Training Set	85%	100%	0%	0%	15%	0%
	Testing Set	83%	92%	13%	8%	4%	0%
Inferential Match							
	Training Set	96%	96%	0%	4%	4%	0%
	Testing Set	90%	92%	10%	8%	0%	0%

Table 6 Results of Experiment 4.

Concept descriptions were obtained using the search procedure. In the case of inferential matching, expert-provided rules were used in combination with flexible matching.

In the first experiment (see Table 3), we have used the events from the training set as a purely factual concept description (Factual Knowledge): the concept was just the disjunction of the training examples. This description is obviously complete and consistent on the training set but has no predictive power for the labor contracts application, i.e. it always produces a `no_match` answer. This happens because in our experiment, as it often happens when dealing with real data, no testing examples were exactly equal to some training event. On the contrary, for the congress voting record application, 2% of the testing examples were equal to some examples in the training set. Therefore even strict matching of the factual knowledge does not always produce a `no_match` answer for the voting record problem.

The factual knowledge was also used with a flexible matching technique, based on a measure of the distance between an event and a concept description (see Section 2.2). Consistency and completeness are achieved in the training set, since flexible matching is only used when either no match or multiple match has occurred. The flexible matching causes some performance improvement for the test set, where 37% of the events are now correctly classified, and we still do not obtain any erroneous classification. This experiment is similar to the simple forms of case-based reasoning described in (Kibler and Aha 1987), where the *one nearest neighbor* (1NN) approach is used to classify events which do not match a single concept description.

1NN approach is also applied to the factual knowledge and results are shown in the bottom part of Table 3. The method used to measure the distance between two instances is same as the method that we use to measure the degree of fit of an event e and a complex (see Section. 2.2). The only difference between flexible matching approach and the 1NN approach is that the scheme used to evaluate the disjunction of complexes. Disjunction of complexes is evaluated as *probabilistic sum* with a threshold in flexible matching approach and as *maximum* in 1NN approach. Flexible matching approach produces a lot of `no_matches`, whereas 1NN produces no `no_match`. This is caused by the threshold (0.5) we set for probabilistic sum. If the distance between two instances is below 0.5, the distance is substituted by 0. As discussed in Section 2.2, probabilistic sum performs worse, when it is applied to a description that consists of many complexes.

In the second experiment (see Table 4), we have used the descriptions learned by AQ15 (Initial Description). Since AQ15 only generates consistent and complete descriptions, classification is 100% correct for the training set, and flexible matching does not affect this performance. For the testing set, the number of correct classifications is still high (80-86%), and flexible matching does not improve the result in this case. This is partly related to the fact that the descriptions generated by AQ are detailed and specify many alternatives, leaving little space for the no_match case (3%). Moreover, the multiple match case was impossible because AQ15 was run with the "disjoint cover" parameter, causing the generated concept descriptions to have disjoint extensions. In general the flexible matching can improve the performance of the initial description on the test set.

If compared with the factual knowledge augmented with 1NN matching, performance of the descriptions generated by AQ is not improved. However, AQ descriptions are order of magnitude simpler, in terms of the number of complexes and selectors. Simplicity is closely related to comprehensibility in the given domain, and allows the performance system to recognize new events more efficiently.

In the third experiment (see Table 5), *base complex* (Michalski et al. 1986) approach is evaluated. The base complex approach is a simple method used in by Michalski et al. (1986) to generate two-tiered concept descriptions. In base complex approach, the BCR is generated by removing all complexes except the most representative one that covers the largest number of training examples. The BCR generated in this approach is very simple and has only one complex for each class. The performance does not drop much, in some cases it is even improved.

The fourth experiment, reported in Table 6, allows us to evaluate empirically the method presented in this paper. We have used the description generated by the search process (Optimized Description), and evaluated its performance both with the flexible matching alone and with the combination of flexible and deductive matching (Inferential Match). For the sake of completeness we also present the performance of the generated descriptions with strict matching, although this would never be used. Strict matching alone yields restricted coverage and poor performance. In fact, the power of the modified description is due to a combination of all three types of matching (strict, flexible and deductive), and all three contribute to the quality measure of a description as computed during the learning process. This represents a new feature of this system, since in earlier work (Bergadano and Giordana, 1989; Michalski et al., 1986) inferential matching is introduced only after the learning phase is completed.

The BCR of the improved descriptions are simpler than the ones generated by AQ15, and, for both data sets, represent the salient characteristics of the concepts being learned. The performance of these descriptions is slightly better if inferential matching is used (3%-6% increase in correct classifications for the testing set, compared to the initial description with flexible matching in Table 4).

The combination of the BCR and the ICI (flexible matching and rules) produces the best results. The description is still simple, although it now includes the ICI rules, and the number of correct classifications is 90-92%. Moreover, some of the examples that were previously recognized by flexible matching or strict matching are now also correctly recognized by the ICI rules, and this might suggest that the description is more robust, and could perform even better on a larger test set. The latter phenomenon also explains the fact that inferential match did not improve performance over flexible matching in the case of Congress data. In our experiment, deductive rules acquired on the training set, when used on the testing set, D-cover events that are also F-covered. Nevertheless, the result must not be underestimated: although performance is the same, inferential matching is preferred over flexible matching. In fact, the examples that are correctly matched through some ICI rule are actually explained by relevant domain knowledge, and not only matched on the basis of a knowledge-independent distance measure. This is important both to accuracy and comprehensibility of the modified description when evaluated with the deductive rules. The quality measure used in the system actually reflects this idea and scores higher when inferential matching is used.

In Table 7, we sum up the results of an experiment that compared the performance of our method with the performance of descriptions obtained from the decision tree building system ASSISTANT (Cestnik et al., 1987). This system was developed from Quinlan's ID3; the basic algorithm was improved to handle incomplete and noisy data, continuous and multivalued attributes. This system also supports tree-pruning mechanisms, which will be discussed in the next section and compared with our method. ASSISTANT was used on the same training sets that produced results reported in Tables 3, 4, 5, and 6.

These results indicate that a two-tiered approach to learning allows a system to learn concepts from a small number of examples, and produce highly accurate descriptions. The performance of different concept descriptions is measured by the percentage of training set events recognized correctly by each description. In Table 7, the Factual Description again denotes the disjunction of training examples, plus the flexible matching function.

The Initial Description is provided by AQ15 and also uses the flexible matching; the missing 3% of the training set represent the `no_match` situation. The third row of the table shows ASSISTANT's performance on the training set. It should be mentioned that the results for ASSISTANT were obtained with decision trees optimized using the tree-pruning process (Cestnik et al., 1987), rather than with the purely inductive learning. The last row shows the performance realized by the Optimized Two-tiered Description.

All these experiments show how a two-tiered learning scheme allows a system to learn concepts from a small number of examples, and produces simpler but still accurate descriptions. The concept meaning is now divided into a cover, generated by the search, a flexible matching procedure, defined a-priori, and a set of deductive rules given by the expert on the basis of misclassified examples selected automatically during the search process.

	Labor (2 classes)	Congress (2 classes)
Factual Description		
Performance (flexible matching) (%correct / % incorrect)	37 / 0	66 / 2
Performance (1-nearest neighbor) (%correct / % incorrect)	77/23	85/ 15
Complexity (#complexes / #selectors)	27 / 432	51 / 969
Initial Description AQ15 (without truncation)		
Performance (%correct / % incorrect)	80 / 17	86 / 14
Complexity (#complexes / #selectors)	11 / 29	10 / 32
Base Complex Description		
Performance (%correct / % incorrect)	83 / 17	85 / 15
Complexity (#complexes / #selectors)	2 / 6	2 / 6
ASSISTANT + PRUNING		
Performance (%correct / % incorrect)	86 / 14	86 / 14
Complexity (#of leaves / tree nodes)	29 / 53	19 / 28
Optimized Description		
Performance (%correct / % incorrect)	90 / 10	92 / 8
Complexity (#complexes / #selectors)	9 / 12	10 / 21

Table 7. Summary of the experimental results.

When the expert provides the typicality of the training events, the method will generate the BCR that covers typical events, and the ICI that is able to classify events with low typicality. When the typicality information is unavailable, as was the case in the experiments described in this section, the system will still produce a two-tiered concept representation. Moreover, this representation may be used to introduce levels of typicality in the set of examples. Examples that are covered explicitly may then be labeled as typical, while examples that are covered inferentially may be categorized as non-typical. Another typicality-related feature of our method its potential noise-resistance. The method will recover from noise, because light complexes and unimportant selectors will be removed (Zhang and Michalski, 1989). More experiments will be needed to verify the impact of typicality on the quality of a concept description, and behavior of the method in the presence of noise.

We conclude this section by showing the concept description obtained by applying AQTT-15 in the labor management domain. The description was obtained by our system using the training set consisting of 18 positive and 9 negative examples of contracts. In Fig. 3 below, we show the complete and consistent initial description produced by AQ15 on this data:

```
[duration ≠ 1] & [wage_incr_yr2 > 3.0] & [holidays ≠
10] v
[wage_incr_yr1 > 4.5%] v
[wage_incr_yr1 > 4.0%] & [wage_incr_yr2 ≠ 2.0% v4.0%]v
[wage_incr_yr1 > 4.5%] & [holidays > 9] v
 [wage_incr_yr1 ≠ 2.0%] & [vacation = above_average] ::>
 acceptable contract

[wage_incr_yr1=2.0% v2.5% v 4.0%] & [holidays=10] &
[vacation=below_average v average] v
[wage_incr_yr1 < 4.5%] & [wage_incr_yr2 = 2.0% v 4.0%]
& [holidays = 10] &
[vacation = below_average v average] v
[duration  =  1]  &  [wage_incr_yr1  <  4.0%]  &
[holidays=9]&
[vacation = below_average v average] v
[wage_incr_yr1=2.0%v2.5% v4.0%] &[wage_incr_yr2=3.0%]&
[vacation=below_average v average]v
[duration = 1] & [wage_incr_yr1=2.0%v2.5% v4.0%]&
[vacation=below_average v average] v
[wage_incr_yr1 = 2.0%] & [wage_incr_yr2 = 3.0%]  ::>
unacceptable contract
```

Fig. 3. Descriptions Generated by AQ15

By running AQTT-15 on the description shown in Fig. 3, a simplified description shown in Fig. 4 has been obtained:

```
[wage_incr_yr2 > 3.0%] v
[wage_incr_yr1  > 4.5%] v
[holidays > 9] v
[vacation = above_average]    ::>  acceptable contract

[wage_incr_yr1 = 2.0% v 2.5% v 4.0%] & [holidays=10]v
[wage_incr_yr2  =  2.0%  v  4.0%]  &  [vacation  =
below_average v average] v
[holidays = 9]  v
[duration = 1] & [wage_incr_yr1 = 2.0% v 2.5% v4.0%]v
[wage_incr_yr2 = 3.0%]  ::>   unacceptable contract
```

Fig. 4. BCR of the Optimized Descriptions of the Acceptable and Unacceptable Labor Management Contracts.

The BCR of the optimized descriptions are much simpler than the ones generated by AQ15, and they represent the most important characteristics of the labor management contracts: a contract is acceptable when it offers a significant wage increase (the first two complexes in Fig. 4), or it offers many holiday days, or the vacation is above average.

The training events that were not correctly classified by the BCR, as it was modified step by step during the search, were analyzed by a domain expert, who provided deductive rules allowing the system to classify almost all the training events (one of them could not be explained by the expert). Fig. 5 shows one of those deductive rules obtained for the ICI of the contract concept.

```
[wage_incr_yr1 < 3.1%] & [wage_incr_yr2 < wage_incr_yr1] ::>
                    unacceptable contract
```

Fig. 5. An ICI Rule from a Two-tiered Description of Unacceptable Labor-Management Contract

The rule addresses the case of a contract with a low wage increase in the first year, and an even lower increase in the second year. In those circumstances, the holiday and vacation offered do not matter: the contract is deemed unacceptable by the union.

7. RELATED WORK

The research presented here improves over the recent work in machine learning that investigates the effects of simplifying concept descriptions, e.g. (Fisher and Schlimmer, 1988; Iba et al., 1988). First, the method described here does not have to experience any loss of coverage as a result of description modification. This is a major difference between experimental results reported in section 6, and the findings of both (Iba et al., 1988) and (Fisher and Schlimmer, 1988). The reason is that in our approach events that lose their strict cover as the result of BCR simplification, may then become covered by the ICI. Moreover, unlike (Fisher and Schlimmer, 1988) and (Iba et al., 1988), this approach is capable of taking into account the typicality of events covered by the simplified description, thus preventing loss of coverage of typical events, whenever typicality information is available.

The experiments of (Fisher and Schlimmer, 1988) in truncating the ID3's decision trees are based on a statistical attribute dependence measure that determines the attributes to be pruned. Because of its statistical character, there is a loss of predictive power when simplifying descriptions learned on small training sets. As the experiments indicate, the approach presented here does not seem to suffer from this problem.

The system developed by (Iba et al. 1988) uses a trade-off measure that is similar to the GDQ measure proposed in this paper. The GDQ measure considers more factors. Besides taking into account the typicality of the instances covered by the description, it considers the type of matching between an instance and a description. Moreover, the simplicity measured by the GDQ depends not only on the number of disjuncts in the description, as in (Iba et al. 1988), but also on the different syntactic features of the terms in the description.

An important difference between the approach presented here and pruning of decision trees (Cestnik et al., 1987; Quinlan, 1987) is lack of constraints on the part of the representation that is truncated when learning a two-tiered concept description. In post-pruning of decision trees, only paths ending in leaves may be truncated, which may improve the efficiency at the expense of the description quality. Moreover, pruning will always specialize one class and generalize the other, while truncation of rules can perform generalization and specialization independently. In (Quinlan, 1987) a method for transforming decision trees into rules and then performing truncation is presented. The method is based on a hill-climbing approach that first truncates selectors and then complexes. No search is performed, only one alternative truncation is tried at every step and the final result may not be the best possible, although the procedure should be faster than in the AQTT-15 system. In the same paper (Quinlan, 1987) other methods for pruning decision trees are also described. Some of these methods require a separate testing set for the simplification phase, others use the same training set that was used for creating the tree. The simplification phase in the AQTT-15 system can be done either with the original training set or with a separate set of examples. None of the presented methods for pruning decision trees involves a search in a space of alternative truncations, i.e., alternative truncations are not backtracking points and are selected irrevocably. As a consequence the best simplification that is possible may never be taken into consideration by these methods. This is also a difference between AQTT-15 and earlier methods described in (Michalski et al, 1986).

Truncation of the BCR, obtained inductively from a small learning set does not affect predictive power if an adequate typicality measure is available. The existence of an adequate ICI further alleviates the problems resulting from induction with few examples.

CN2 inductive algorithm (Clark and Niblett, 1989) uses a heuristic function to terminate search during rule construction. The heuristic is based on an estimate of the noise present in the data. Such pruning of the search space of inductive hypotheses results in rules that may not classify all the training examples correctly, but that perform well on testing data. CN2 can be viewed as an induction algorithm that includes pre-truncation, while the algorithm reported here is based on post-truncation. CN2 applies truncation during rule generation and AQTT-15 applies truncation after the rule generation. The advantage of pre-truncation is efficiency of the learning process, but irrelevant selectors and redundant complexes generated during learning are not removed.

The problem of defining and using the typicality of examples has been considered in the past both in machine learning and cognitive science. Negative examples of low typicality are referred to as *near misses* in Winston's system (Winston, 1975). Such examples, that have to be labeled by the user as near misses, are used in Winston's system to delineate the borders of a concept. (Michalski and Larson, 78) introduced the idea of an outstanding representative of a concept. The concept of prototypical examples has been also studied by (Smith and Medin 1981) and by (Rosch and Mervis 1975).

To summarize, there are four major differences between the work presented here and related research described in the literature. First, the above method may not experience a loss of coverage although it still yields a simpler description with improved predictive power. Second, it simplifies the description by performing independently both generalization and specialization. Third, any part of the description may be truncated in the simplification process. Finally, the method takes into account the typicality of the examples and a general description quality measure is used.

From a more general standpoint, it seems interesting to situate the method introduced here in the spectrum of existing machine learning approaches. The methods that we would like to relate to are:

simple inductive techniques (concepts are represented on one level only), and case-based reasoning. Table 8 below describes these methods, as well as the two-tiered approach in terms of the type of concept representation, and the kind of matching applied for classification.

	Simple Induction	Case-based	Two-tiered
Representation	General	Specific	General
Matching	Precise	Inferential	Inferential

Table 8. Comparison of Two-tiered Approach with Simple Inductive and Case-based Methods

8. CONCLUSIONS AND OPEN PROBLEMS

The paper describes a method of learning two-tiered concept descriptions. The method is based on transforming an initial Base Concept Representation. The transformed BCR covers groups of events characterized by high typicality. It is also syntactically simpler, and therefore more comprehensible than the initial BCR. A more complete coverage of the events from the learning set by the whole two-tiered description is achieved through inference. The method presented in this paper relies not only on the probabilistic inference, implemented as a flexible matching function in (Michalski et al., 1986). It uses also a rule base for deductive inference. Deductive inference has the additional advantage of explaining why a given event is to be included (or excluded) from the cover.

Transformations of BCR are implemented as truncations of the cover. The cover is provided by AQ15 in a standard, disjunctive normal form. The truncations either specialize the description (complex removal, referent shrinking), or generalize it (selector removal, referent extension). A search process, guided by a quality measure, is used to obtain a "good" description. The measure takes into account not only the explicit part of the description, but also the implicit one.

The experimental results that we obtained confirm the hypothesis that two-tiered descriptions can be simpler, more accurate and easier to understand. The ICI used in the experiments included a flexible matching function and a set of logical rules. The performance of the descriptions produced by the search process on the test set is influenced by the use of the inferential matching. This is due to the fact that ICI is used during learning, in order to choose and modify the best descriptions. This property represents an important difference between the presented system and previous approaches, that tend to apply flexible matching only after the learning process is completed.

There is a number of advances and differences of the method presented here, compared to previous work (Michalski et al, 1986). That earlier approach produced some preliminary results in which flexible matching function was applied during the testing phase. The same research investigated the effect of truncating concept descriptions. In the system presented here, though, the flexible matching function is augmented by a set of rules defining how to extend or modify a concept description at the "knowledge level" (Dietterich 1986), by describing symbolically its possible transformations. Besides complex removal, the new operators of selector removal and referent modification are introduced. In the TRUNC approach (Michalski et al., 1986) truncations of the description were applied manually, with a limited number of tries. This is in contrast with AQTT-15, which is based on the automatic search for better two-tiered descriptions.

An important issue for future research and improvements of the implemented system is the integration of the search procedure with the inductive learning system used to generate the initial description (AQ). The first step in this direction is being experimented with: it allows the two systems to share the same heuristics and the same measure of quality. Further progress is related to the possibility of obtaining partially incomplete and inconsistent description also during the generation of the initial description. More experimentation is also needed in order to evaluate more precisely the performance of the implemented system.

The relationship of the work presented here with analytical learning methods has at least two aspects. First, we view the generation of a two-tiered description as a knowledge transformation process. If this process is purely deductive, one obtains the standard EBL method (Mitchell et al. 1986). In general, the transformation involved may not be truth-preserving, as is the case in our system. The transformed knowledge is expected to be more accurate and more efficient, according to our description quality measure GDQ. This measure corresponds to the notion of operationality in EBL. Second, the problem of automatic acquisition of the ICI rules has to be investigated. These rules can either be inherited from higher level concepts or provided by the expert. The methods developed in Explanation-Based Learning will provide a good starting point, when the expert is able to specify classification knowledge which is correct but not operationally effective. If the knowledge supplied by the expert is not correct or is too specific it can be improved by inductive methods. It has to be observed, however, that since the events to be explained are usually exceptions, the knowledge necessary to explain them may be lacking from the system.

The problem of learning second tier rules has to be addressed in future. One approach, currently under development (Plante and Matwin 1990), learns ICI rules using chunking techniques in an environment in which multiple explanations for both positive and negative training events are provided.

Another form of knowledge level learning is constructive induction (Michalski 1983). In general, this approach produces descriptions that are easier to understand, and capture the salient features of the concept. The effects of this type of induction are even more relevant for our method. Constructive induction folds several disjuncts into a single one. Moreover, the use of constructive rules will usually merge relevant conjunctive descriptions of the concept, regardless of how many examples they cover. This feature of constructive induction may prevent removal of relevant complexes.

The system in its current form does not address the problems of dynamically emerging hierarchies of concepts. The system only learns one concept at a time, and concepts do not change or split as new examples become available.

Another open issue is the ability of the system to self-reorganize. The distribution of knowledge between the BCR and the ICI will be determined by the performance of the system on large testing sets. If it turns out, e.g., that some ICI rules are used very often, then these rules could be compiled into explicit BCR assertions. It seems, therefore, that in concept representation one can trade one parameter against the other, within certain limits. This interesting research problem merits further investigation.

Finally the system is being experimented also with structural descriptions, and in this case the initial description to be simplified is provided by INDUCE, rather than by AQ. The search procedure is essentially the same, but the computation of the GDQ and of the search heuristics is less efficient. A different kind of flexible matching needs to be defined for structural concept descriptions. This paper has been focusing on attributional descriptions, and the experiments that were presented do not deal with relationships among objects in the examples.

ACKNOWLEDGEMENTS

The authors thank G. Thornburg, K. Kaufman, A. Giordana, F. Oppacher and H. de Garis for their useful comments. Zbig Koperczak's help in acquiring the test data is kindly acknowledged.

This research was done in the Artificial Intelligence Center of George Mason University. Research activities of the Center are supported in part by the Defence Advanced Research Projects Agency under grant, administered by the Office of Naval Research, No. N00014-87-K-0874, in part by the Office of Naval Research under grant No. N00014-88-K-0226, and in part by the Office of Naval Research under grant No. N00014-88-K-0397. The first author was supported in part by the Italian Ministry of Education (ASSI) project, and the second author was supported in part by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Bareiss, R., "An Exemplar-based Knowledge Acquisition", Academic Press, 1989.

Bergadano, F., Giordana, A., "Pattern Classification: An Approximate Reasoning Framework", International Journal of Intelligent Systems, 1989.

Bergadano, F., Matwin, S., Michalski, R. S., Zhang, J., "Measuring Quality of Concept Descriptions", Proc. Third European Working Sessions on Learning, pp. 1-14, Glasgow, 1988.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., Freeman, D., "AutoClass: A Bayesian Classification System", Procs. of the Fifth Int'l. Conf. On Machine Learning, Ann Arbor, pp. 54-64 (1988).

Cestnik, B., Kononenko, I., Bratko, I., "ASSISTANT 86: A Knowledge-elicitation Tool for Sophisticated Users", Procs. of the 2nd European Workshop on Learning, pp. 31-45 (1987).

Clark, P. Niblett, T., "The CN2 Induction Algorithm", Machine Learning, vol. 3, No.4, , pp. 261-183, 1989.

Collins, A. M., Quillian, M. R., "Experiments on Semantic Memory and Language Comprehension" in Cognition, Learning and Memory, L. W. Gregg ed., John Wiley, 1972.

DeJong, G., Mooney, R., "Explanation-based Learning: An Alternative View", Machine Learning, vol. 1, No. 2, 1986.

Dietterich, T., "Learning at the Knowledge Level", Machine Learning, vol. 1, No. 3, pp. 287-315, 1986.

Dietterich., T., Flann, N., "An Inductive Approach to Solving the Imperfect Theory Problem", Procs. of the Explanation-based Learning Workshop, Stanford University, pp. 42-46, 1988.

Fisher, D. H., Schlimmer, J. C., "Concept Simplification and Prediction Accuracy", Procs. of the Fifth Int'l. Conf. On Machine Learning, Ann Arbor, pp. 22-28 (1988).

Hammond, K., "Case-based Planning: Viewing Planning as a Memory Task", Academic Press, 1989.

Iba, W., Wogulis, J., Langley, P., "Trading off Simplicity and Coverage in Incremental Concept Learning", Procs. of the Fifth Int'l. Conf. On Machine Learning, Ann Arbor, pp. 73-79 (1988).

Kedar-Cabelli, S. T., McCarty, L. T., "Explanation-based Generalization as Resolution Theorem Proving", Procs. of the 4th Int. Workshop on Machine Learning, Irvine, 1987.

Kibler, D., Aha, D., "Learning Representative Exemplars of Concepts", Procs. of the 4th Int. Workshop on Machine Learning, Irvine, 1987.

Kolodner, J., ed., Proceedings of the Case-based Reasoning Workshop, DARPA, Clearwater Beach, FL, 1988.

Lebowitz, M., "Experiments with Incremental Concept Formation: UNIMEM", Machine Learning, vol. 2, No. 2, 1987.

Michalski, R.S., Larson, J. B., "Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: the Underlying Methodology and the Description of Programs ESEL and AQ11", TR 867, Dept. of Computer Sci., UIUC, 1978.

Michalski, R.S., "A Theory and Methodology of Inductive Learning", Chapter in the book "Machine Learning, an Artificial Intelligence Approach", Michalski, R. S., Carbonell, J. G., Mitchell, T. M. (Eds.), Tioga Pub. Co., Palo Alto, Ca, 1983.

Michalski, R. S., Mozetic, I., Hong, J., Lavrac, N., "The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains", Proc. 5th AAAI, pp. 1041-1045, 1986.

Michalski, R. S., "Two-Tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness" , Chapter in the Book "Similarity and Analogy", Stella Vosniadou and A. Orton, (Eds), 1988.

Michalski, R. S., Ko, H., "On the Nature of Explanation, or Why Did the Wine Bottle Shatter", AAAI Symposium: Explanation-Based Learning, Stanford University, pp. 12-16, 1988.

Michalski, R. S., "How to Learn Imprecise concept: A Method Employing a Two-Tiered Representation for Learning", Procs. of the Fourth International Workshop on Machine Learning, Irvine, CA, pp. 50-58, 1987.

Mooney, R., Ourston, D., "Induction Over the Unexplained: Integrated Learning of Concepts with Both Explainable and Conventional Aspects", Procs. of 6th Int'l WoÂkshop on Machine Learning, Ithaca, NY, pp. 5-7, 1989.

Minsky, M., "A Framework for Representing Knowledge", in The Psychology of Computer Vision, ed. P. Winston, (1975)

Mitchell, T. M., Keller, R., Kedar-Cabelli, S., "Explanation-based Generalization: a Unifying view," Machine Learning Journal, vol. 1, No. 1, pp. 11-46, 1986.

Mitchell, T. M., "Version Spaces: An Approach to Concept Learning", Ph. D. dissertation, Stanford University, December 1977.

Plante, B., Matwin, S., "Learning Second Tier Rules by Chunking of Multiple Expalnations", Research Report, Department of Computer Science, University of Ottawa, 1990.

Prieditis, A. E. and Mostow, J., "PROLEARN: Towards A Prolog Interpreter that Learns," Proc. of IJCAI 87, Milan, pp. 494-498, 1987.

Quinlan, J. R., "Simplifying decision trees", Int. Journal of Man-Machine Studies, vol. 27, pp. 221-234, 1987.

Robinson J. A., Sibert E. E., "LOGLISP: An Alternative to Prolog", Machine Intelligence, vol. 10, Hayes, J. E. and Michie, D. (Eds.), pp. 399-419, 1982.

Rosch, E., and Mervis, C. B., "Family Resemblances: Studies in the Internal Structure of Categories", Cognitive Psychology, vol. 7, pp. 573-605, 1975.

Smith, E. E., Medin, D. L., "Categories and Concepts", Harvard University Press, 1981.

Sowa, J. F., "Conceptual Structures", Addison Wesley, 1984.

Sturt, E., "Computerized Construction in Fortran of a Discriminant Function for Categorical Data", Applied Statistics, vol. 30, pp. 213-222, 1981.

Watanabe, S., "Knowing and Guessing, a Formal and Quantitative Study",Wiley Pub. Co., 1969.

Weber, S., "A General Concept of Fuzzy Connectives, Negations and Implications based on t-norms and t-conorms", Fuzzy Sets and Systems, vol. 11, pp. 115-134, 1983.

Winston, P. H., "Learning Structural Descriptions from Examples", in "The Psychology of Computer Vision", P. Winston, ed., McGraw-Hill, 1975

Zadeh, L. A., "Fuzzy Logic and its Applications to Approximate Reasoning", Information Processing, North Holland, pp. 591-594, 1974.

Zhang, J. and Michalski, R. S., "Rule Optimization via SG-Trunc Method", Proc. Fourth European Working Sessions on Learning, 1989.