Knowledge Acquisition
via
Knowledge Integration

Pavel B. Brazdil
Luís Torgo

LIACC
Laboratory of AI and Computer Science

University of Porto
Rua Campo Alegre, 823 - 2º
4100 Porto, Portugal
ltorgo, pbrazdil@ncc.up.pt

Abstract. In this paper we are concerned with the problem of *acquiring knowledge* by *integration*. Our aim is to construct an integrated knowledge base from several separate sources. The need to merge knowledge bases can arise, for example, when knowledge bases are acquired independently from interactions with several domain experts. As opinions of different domain experts may differ, the knowledge bases constructed in this way will normally differ too. A similar problem can also arise whenever separate knowledge bases are generated by learning algorithms. The objective of *integration* is to construct *one* system that exploits *all* the knowledge that is available and has a good performance. The aim of this paper is to discuss the methodology of knowledge integration, describe the implemented system (INTEG.3), and present some concrete results which demonstrate the advantages of this method.

1. Introduction

The areas of *knowledge acquisition* (KA) and *machine learning* (ML) have until recently existed as two rather distinct areas, despite the fact that both areas share quite similar concerns. Both areas are attempting to construct systems that model certain phenomena in the real world so as to aid us in decision making.

Most of the research in KA presupposes that the model is created by a person (an `expert´) first. The objective of *expertise transfer* (ET) tools is to help the user to transfer this expertise into a computer knowledge base first. In *machine learning*, on the other hand, the system is usually supplied with *data* which is stored in the knowledge base. The process of *model creation* is left to the system, that is, to the appropriate empirical inductive tool.

As Gains (1989) has pointed out each approach has its own difficulties. Experts do often provide a mixture of relevant and irrelevant or erroneous information. The objective of current expertise transfer tools, such as Aquinas, KSS0 or KADS (Wielinga & Breuker, 1986) is to help the user to clean up the information provided. The disadvantages of a pure machine learning approach are also quite obvious. The system is too dependent on the data that is provided from outside. This data need not necessarily be complete. Some important cases may simply be missing. Also, a part of the data may be corrupted. The data may contain incorrect information.

Some people have expressed the view that both methodologies should be combined (Gaines, 1989; Boose et al., 1989), but not many concrete proposals were put foward so far. Aquinas, for example, was capable of presenting to the user opinions of multiple experts, but the user was responsible for weighing opinions and selecting the solutions to be followed. Also, as Aquinas used *repertory grids* for representation of concepts, the language used for representing knowledge is somewhat limited. As Boose et al. (1989) point out, further facilities are needed for elicitation and analysis of knowledge from multiple experts, resolving potential conflicts (e.g. by negotiation) and knowledge base merging.

In this paper we describe a method for merging several separate theories (knowledge bases)[1]. We assume that, in general, these will have been generated in different ways. That is, some theories can be obtained by querying an expert and transcribing his knowledge in the form of rules. Others can be obtained on the basis of data, using inductive learning tools (see Fig. 1). System INTEG.3 is capable of analysing the individual theories and its rules, and selecting (or marking) some to be included in the integrated theory. These selected rules determine the decision of the overall system.

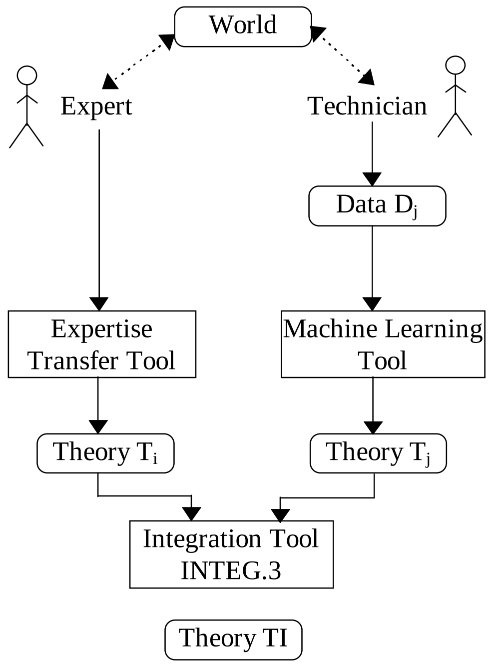

Fig. 1 Possible Use of Integration Tools

Most of the experiments we have performed were concerned with a medical domain[2]. All the theories that were used by INTEG.3 were generated by inductive learning tools. The theories differed from one another in various ways, and it was not a priori clear which theory was right. In the experiments the integrated theory had significantly better performance than the original theories. The experiments confirmed our belief that knowledge integration can play an important role in the process of knowledge acquisition.

---

[1] In this paper the term "theory" is used interchangeably with the term "knowledge base". Both terms have a rather restricted meaning here. They are used to identify a collection of rules (e.g. Horn clauses).

[2] The tests referred to here were performed on lymphography data (obtained from JSI, Ljubljana) which contained 4 possible classifications. This may not be exactly the same data as the one used by Gams (1989) which contained 9 possible classifications.

Knowledge Integration and Incremental Learning

Knowledge integration is concerned with issues that are related to those in *incremental learning* systems, such as ID4 (Schlimmer and Fisher, 1988), ID5 (Utgoff, 1988), AQ16 (Janikow, 1989), for example. Both knowledge integration and incremental learning attempt to construct a theory that explains best the given data. There are some important differences between the two approaches, however.

When we talk about incremental learning, usually it is assumed that only *one system* is constructing theories by employing some incremental version of a given learning algorithm. In consequence all the data is actually analyzed by this system at one time or another. Knowledge integration, on the other hand, involves *several systems* all of which try to construct their own theories on the basis of their *own* experience. Although knowledge integration may require some common experience in the process of constructing an integrated theory, we are not in favour of simply transferring all the data to one system. People do not use only "raw data" when they communicate with others about certain world phenomena.

If we let the systems communicate theories, instead of data, some learning effort will be saved, but we can expect that some information will be lost in the process. We could thus expect that the results obtained by knowledge integration may not succeed the results obtained by incremental learning systems. Our experiments have indicated the contrary. The performance of the integrated theory was better than the performance of the incremental system (see Section 2.4 for more details).

Knowledge integration has, however, one additional advantage over incremental systems which are unable to improve upon theory provided by the user. Incremental learning systems keep various internal data structures and statistical measures in the memory (e.g. informativity of various attributes etc.). These enable the system to update its theory when new data becomes available. If these are not supplied by the user (or alternatively, if the user does not provide all the examples he has seen), the incremental systems can do little to improve the given theory.

2. Method of Knowledge Integration

In this section we will describe the method of knowledge integration in more detail. Basically, the process involves a preparatory phase in which a group of systems (agents) is selected and organized into a group that can function together. The group can include systems S_1-S_n that are capable of constructing theories from data, responding to external queries and communicating with one another. Here we shall assume that this has already been done and that the system organization is fixed[3] .

Having defined the organization of the multi-agent system, it is necessary to determine the overall objectives. That is, for example, one can define which concepts are to be acquired and/ or describe in some way the required performance. This is important if we want the multi-agent system to decide when to stop altering its theories. Here we will follow what has been done in the past and let the user control this issue.

Consequently, here we will be concerned with, basically, the following three phases:

(1) Generation of independent theories (by consultation or inductive learning),

---

[3] In general systems can be given the ability tp recognize themselves. These are some of the concerns of Distributed Artificial Intelligence (see e.g. (Bond and Gasser, 1988)), but they are outside the scope of this paper.

(2) Competitive characterisation of the system´s theories,
(3) Construction of the integrated theory.

In phase (1) the systems S_1-S_n work in an independent manner, and as a result produce theories T_1-T_n. Each system involved constructs its *own theory* on the basis of its own experience. Here S_j can represent either human, or an inductive learning tool. In either case S_j will produce theory T_j .

In phase (2) the individual theories are characterized using tests. Without loss of generality let us assume that this is actually controlled by some agent SI. This agent poses a query to all the agents involved, waits for the answers and then proceeds with the next query. Any of the systems S_1-S_n could act as SI. The subsystem responsible for characterization of theories is referred to as INTEG.3.

Phase (3) is dedicated to the issue of constructing one integrated theory (TI) on the bases of the results obtained in phase (2). This task is also done by INTEG.3. The three phases mentioned could be followed by two additional ones:

(4) Adoption of integrated theory by one (or more) systems.
(5) Check whether the process should continue, and if so, go to (1).

In this paper we will be concerned mainly with the phases (1)-(3). The issue of hoe one could construct a `closed loop system´, capable of taking the integrated theory and using it as input in further learning will be discussed in a future paper. The next section describes phases (1)-(3) in more detail.

2.1 Generation of Individual Theories

Use of Inductive Learning Tools

The decision concerning which system we should choose to generate individual theories is not really too important in the context of we want to prove. Here we require only that the system(s) are capable of generating theories that perform reasonably well on tests. As we had earlier reimplemented ID3- and AQ-like systems, we decided to use these as the basic inductive engines in our set-up.

The reimplementation of ID3 based on earlier work (e.g. Quinlan (1986), Cestnik et al. (1987), Clark and Niblett(1987; 1989)) will be referred to as ITL1 (*Inductive Tree Learning System*). The decision tree generated by this system is automatically converted into a rule form which we find more amenable for further manipulation.

The inductive rule learning system IRL1 is an incremental learning program, that was partially inspired by CN2 (Clark and Niblett, 1989). This system incrementally updates the existing rules either using generalization and/ or specialization. More details about this system will be made available in (Torgo, 1990).

Different theories needed by the knowledge integration system (INTEG.3) were generated by the inductive learning systems (ITL1 and IRL1) in a series of *independent learning tasks*. In each task the inductive learning system generated a theory (consisting of a set of rules) on the basis of its own data. Let us see how the data was prepared for each experiment.

From the total number of cases available (D), 30% were separated out by a random process and reserved for the final tests of the integrated theory. Let us call this test set, set DT. If we disregard the cases in this set, 70% of the original data can be available for the creation of alternative theories (see Fig. 2). These are constructed with the help of different inductive learning systems. In most of our experiments we have generated four theories. Some were generated by ITL1 and others by IRL1.

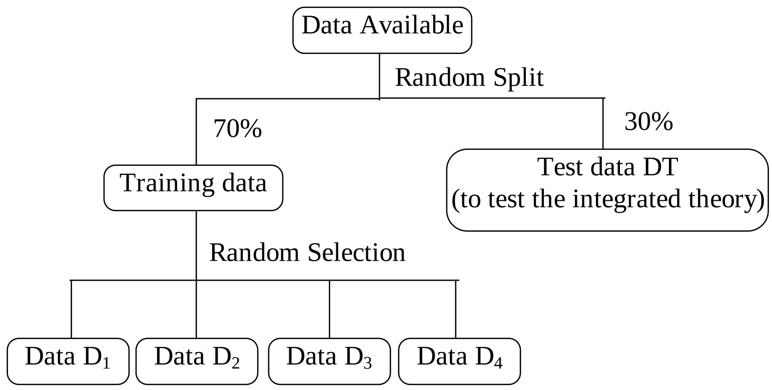

Fig. 2 Generation of Various Training Sets

In each experiment some cases were selected at random from the pool of data available (D-DT). Let us call this set D_i. Set D_i was then supplied to the inductive learning system to generate theory T_1. This process was then repeated to generate the next theory (see Fig. 3). As sets D_1. .D_n were selected at random from the same pool of examples, the sets mentioned could have some cases identical to those used by others.

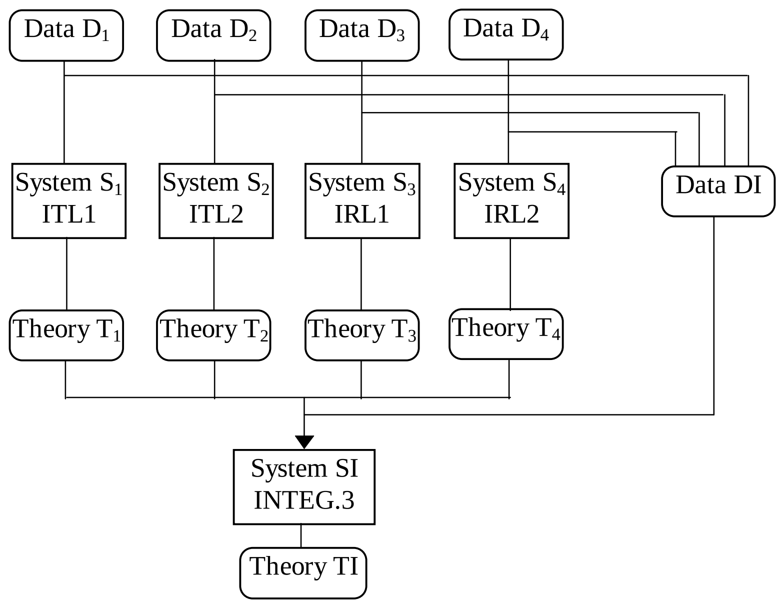

Fig. 3 The Set-up for Knowledge Integration Experiments

2.2 Competitive Characterization of Theories

After the theories have been constructed by the individual systems they are characterized on the basis of integration tests for which we need some data. Let us refer to this data as the set DI. The easiest way to obtain this set is to take the union of the data sets D_1. .D_n. This is the approach we have actually adopted. In general, however, the set DI may consist of any other representative sample of data.

The results of tests are both qualitative and quantitative. The objective of the *qualitative* characterization is to provide detailed information about relative benefits of individual rules or theories to the integration system. This information takes the form of lists of cases covered by a particular theory or rule. The objective of the *quantitative* characterization is to estimate the overall *accuracy* of the individual theory (or rule).

Qualitative Characterization of Theories

Qualitative characterization of *theories* is similar to qualitative characterization of *rules*, and so in the following we shall simply speak about qualitative characterization of rules.

Qualitative characterization of a particular rule R consists of two lists. The first one mentions all the test cases that were *correctly* covered by this rule. In other words, this list refers to the *positive examples* covered by the rule. The second list mentions all the cases that were *incorrectly* covered by the rule. This list represents the *negative examples* covered by the rule in question. The list mentioned do not need to contain complete descriptions of each case. For our purposes it is sufficient to store only case identifiers (indexes) that uniquely identify each case.

Quantitative Characterization of Theories

Quantitative characterization of a given theory is done using an estimate of accuracy. These are calculated on the basis of test results on the test set DI.

The tests are done in a usual manner by comparing the classification predicted by the theory T_i with the classification stored together with the case. This comparison enables us to decide whether the particular theory classified the case correctly or not. The classification errors caused by *misclassification* are sometimes referred to as errors of *commission*. Errors of *omission* arise whenever an expert or a system fails to classify some case, that is, when no classification is actually predicted.

The accuracy of theory *T* is estimated using the following formula:

$$AC_T = C_T / (C_T + EN_T + ER_T)$$
$$= 1 - (EN_T + ER_T) / (C_T + EN_T + ER_T)$$

where C_T represents the number of correctly classified cases, ER_T the number of misclassifications, and EN_T errors of omission. In effect, the coefficient Q_i represents a ration of correctly classified cases (C_T) to all cases used in testing ($C_T + EN_T + ER_T$).

Quantitative Characterization of Rules

Quality of individual rules is estimated in a heuristic manner, using the expression

$$Q_{T,R} = AC_{T,R} * \text{Estimate\_of\_coverage}$$

where $AC_{T,R}$ represents the *accuracy* of rule R and *Estimate_of_coverage* is a number between 0 and 1 that guides the system to prefer rules with relatively high covergage of a class. Its value is calculated as follows:

$$\text{Estimate\_of\_coverage} = \exp(C_{T,R} / N_{C,R} - 1)$$

The value $C_{T,R}$ represents the number of correct classifications. The value of $N_{C,R}$ is obtained as follows. The consequent of rule *R* determines which class (C) the rule is concerned with. Then $N_{C,R}$ represents the total number of examples of that class (C). The accuracy of individual rules is estimated in a similar way as accuracy of theories using:

$$AC_{T,R}= C_{T,R}/ (ER_T + C_{T,R})$$

Here $AC_{T,R}$ represents the accuracy of rule *R* belonging to theory *T* and ER_T represents the number of errors. We notice that this formula takes into account errors of commission, but not errors of omission. The reason is that whenever several rules are associated with a particular class, and no rule is activated, it is difficult to decide which rule is at fault. This is why we have decided not to include these errors in our estimates.

Let us consider an example. In our system, the assertion,

rule(1,5 metastase <----earlyuptakein = no ^ defectinnode = lac_central, 15,3).

for example, represents rule 5 of theory T_1. The numbers 15 and 3 represent number of correctly classified cases ($C_{1,5}$) and number of misclassifications ($ER_{1,5}$). The accuracy of this rule is thus

$$AC_{1,5}= C_{1,5}/ (ER_{1,5} + C_{1,5}) = 15/ (3 + 15)= 83\%$$

Supposing we have 30 examples of metastase then $Q_{1,5}$ would be calculated using:

$$Q_{1,5} = 0.83 * \exp(15/30 - 1)= 0.83 * 0.61$$

Both qualitative and quantitative characterization of theories are used in the construction of the integrated theory.

2.3 Generation of the Integrated Theory

The integrated theory TI is constructed on the basis of the candidate set TC. Initially, set TC contains all the rules of the individual theories $T_1..T_n$. The objective is to select some rules from TC and transfer them into TI so as to achieve good performance (accuracy). The method relies on the qualitative and quantitative characterization of rules and includes the following steps:

(1) Order rules in the candidate set according to their estimates of quality.
(2) Select the rule R with the best quality and include it in TI.
(3) Mark the cases covered by R.
(4) Recalculate the quality estimates of rules excluding the marked cases.
(5) Go back to (1).

The process of adding new rules to TI terminates when the accuracy of `best rule´in TC falls below a certain threshold. In our experiments the limit was set to a rather low value (1%). Even this low value, however, prevented the inclusion of many rules in the integrated theory. These were the rules that covered no examples (all the examples have already been covered).

2.4 Results

We have conducted a series of experiments whose purpose was to compare the performance of the integrated theory with the performance of the original theories. The systems were asked to construct theories on the basis of 5, 10, ...30 training examples. Each theory was obtained by an inductive process on the basis of a given number of examples which were drawn from a given pool by a random process. In order to exclude the possibility of fortuitous results each experiment was repeated ten times. The performance of system S_1, for example, obtained on the basis of 15 training examples was assessed by calculating the mean performance of 10 theories generated by this system. The mean performance in this case was 57.3% (the standard deviation was 10.4%). The results of our experiments are shown in the following table.

	5	10	15	20	25	30ex.
S_1	39.3	51.6	57.3	47.5	58.2	60.9
	(17.2)	(14.2)	(10.4)	(9.4)	(6.7)	(7.8)
S_2	46.8	55.4	55.5	59.1	62.3	56.8
	(16.5)	(10.1)	(14.9)	(7.4)	(5.5)	(8.0)
S_3	52.9	54.8	55.2	72.9	57.1	68.1
	(17.5)	(15.9)	(7.1)	(10.3)	(11.5)	(3.8)
S_4	54.5	57.0	67.5	66.6	73.0	75.7
	(16.1)	(17.0)	(10.8)	(8.3)	(10.4)	(4.8)
S_M	48.4	54.7	58.9	61.5	65.1	65.4
	(9.6)	(9.1)	(5.2)	(4.7)	(4.1)	(3.2)
T_1	70.4	73.6	80.0	82.7	83.9	85.4
	(12.9)	(10.0)	(7.5)	(6.6)	(6.3)	(3.9)
Gain	22.1	18.9	21.1	21.2	18.8	20.1
	(8.8)	(8.0)	(5.1)	(5.9)	(6.9)	(4.7)

Table 1. Performance of the Integrated Theory TI and the Individual Systems (S_1 - S_4).

The first four rows of the table show the performance figures of the individual systems (S_1 - S_4). Systems S_1,S_2 use ITL1 inductive learning method while systems S_3,S_4 use IRL1. All the figures shown are *means* obtained on the basis of ten independent experiments. The standard deviation is shown in brackets. The values of S_M represent the mean value calculated as follows: $S_M = (S_1 + S_2 + S_3 + S_4)/ 4$. This value is useful when trying to assess the performance gain of the integrated theory TI. The last row shows the gain (TI - S_M). As can be seen the performance gains were around 20%.

Fig. 4 shows some of the results in a graphical form. As we can see system S_1 follows a learning curve. It is interesting to note that the performance tends to fluctuate less as more examples are given. That is, the standard deviation decreases with the number of examples. The performance of the integrated theory exceeds the performance of S_1 (and the systems). The performance gains obtained were really quite surprising. They were around 20%. The integrated theory was also quite "stable" in the sense that its performance did not fluctuate as much as the performance of the individual systems.

An interesting question is whether one particular system, say S_1, could obtain a similar performance as the integrated theory (TI), had it been supplied with all the data. Here we shall call this system S_1+. We have conducted a separate experiment to compare the performance. The results are shown in Table 2. We have found it rather surprising that the integrated theory had better performance than system S_1+.

	5	10	15	20	25	30ex.
S_1+	50.0 (11.9)	58.2 (11.4)	63.7 (9.1)	66.6 (9.1)	66.8 (10.3)	73.4 (7.4)
TI	70.4 (12.9)	73.6 (10.0)	80.0 (7.5)	82.7 (6.6)	83.9 (6.3)	85.4 (3.9)
Gain	19.6	15.4	16.3	16.1	17.1	12.0

Table 2. Performance of the Integrated Theory and System S_1+.

The first row of this table shows the performance of system S_1 that has been additionally supplied with the data of systems S_2-S_4. The extra data helped S_1 to improve its performance, but the performance of the integrated theory (TI) was better.

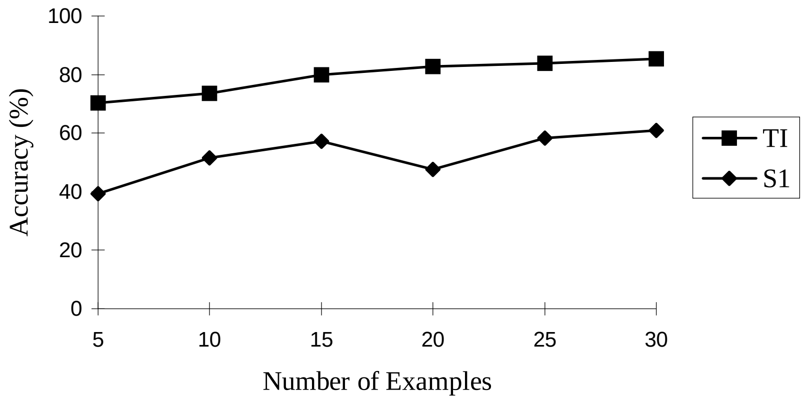

Fig. 4. Performance of the Integrated Theory TI and System S_1. In this figure the performance of the integrated theory (TI) is contrasted with the performance of systems S_1. Standard deviations are indicated by vertical bars. The graph is based on the data shown in Table 1.

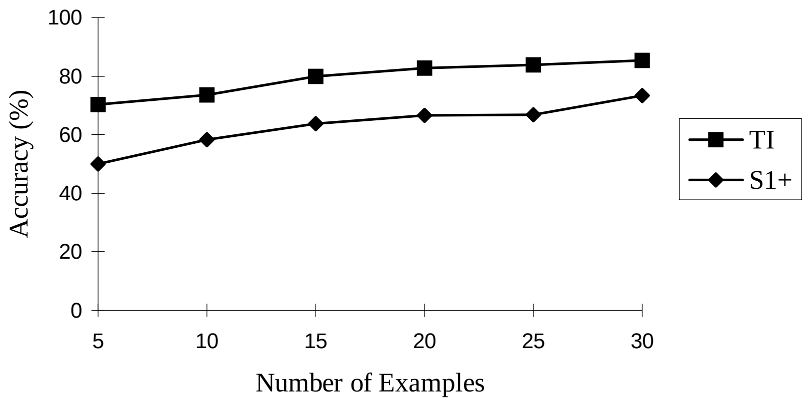

Fig. 5. Performance of the Integrated Theory and System S_1+.

The results of these experiments have exceeded our expectations. We have expected the integrated theory to perform well, but the performance gains were really higher than we have expected. Had there been anything wrong with our experiments?

It is conceivable (although unlikely) that the present versions of ITL1 and IRL1 have a suboptimal performance. But this would not be an argument against integration. System INTEG.3 tries to make the best out of the theory it gets, irrespective of whether they are optimal or not. INTEG.3 checks them out and produces an integrated theory that, as we have seen, performs generally well.

The experiments confirmed our belief that it is worth pursuing this issue further. In the next section we discuss the relationship of this method to other work. We also describe some of the shortcomings of our approach and possible extensions that could be made in future.

3. Discussion

Integration is a Form of Learning

The process of constructing a theory from several individual theories can be regarded as a *form of learning*. The process can be regarded as *learning*, because the system´s internal representation becomes reformulated and this leads to an improvement of performance. We notice that the reformulations are rather simple. They involve inclusion (or non inclusion) of rules in the integrated theory and modification of coefficients affecting their selection. Consequently, we use the term *weak* form of learning to distinguish it from other more complex forms of theory reformulation which could also involve modifications of individual rules. Despite its simplicity, however, this method leads to noticeable improvements in performance. More work can, of course, be done in this area. In the following sections, we will discuss some possible extensions of this work.

Knowledge Integration and Redundant Knowledge

Gams (1989) has argued that redundant knowledge is essential for superior behaviour of learning systems and his measurements support this claim. Gams further justifies this approach as follows: "*It is commonly accepted that cross checking of several knowledge sources is generally better than using one source of knowledge alone*."

It is conceivable that INTEG.3 achieves higher performance than expected thanks to some kind of in-built redundancy mechanism. It is worth pointing out, however, that Gam´s approach is different from ours. Gams attempts to represent the redundant knowledge in an *explicit* manner. Although INTEG.3 also exploits several theories in the process of constructing an integrated theory, there is no attempt to represent redundant knowledge within the system. The integrated theory has a similar structure as the original theories. Also, it is not much bigger in size than any of the original theories. Thanks to the fact that the integrated theory has relatively simple form, it can be used as input in further learning.

Analytic Versus Empirical Approaches

Our system could be extended to perform an analysis of the rules that have been obtained. The system could try to *merge* the existing rules and *reformulate* them so as to avoid possible conflicts and overlaps, as was suggested, for example, by Steels and Van de Welde (1989). Our system constructs the integrated theory solely on the basis of empirical evidence. However, empirical approaches are quite powerful.

Consider, for example, the problem of determining whether one rule *subsumes* another. This is a difficult task in general. The failure to conclude the proof within a given time does not mean that proof will not possibly succeed later. If on the other hand, one relies on an empirical approach, the task is somewhat easier. INTEG.3, for example, will analyze the given cases to get a rule that covers some of them. Then it will consider the cases that have not been covered yet, and so on. The chances that INTEG.3 would generate rules that subsume others seem to be small. However, if the domain were very noisy, it could perhaps happen that inconsistent rules would appear in the integrated theory. Of course, preference would always be given to the rule that is qualitatively better, so the problem would not really affect much of the system´s function. But one could argue that whenever the data is noisy, it is wrong to try to generate perfect rules that `hide the problem under the carpet´.

Knowledge Integration and Theory Revision

The BLIP system (Emde, 89) is capable of representing several *competing theories*. Emde points out that the decision whether some fact is consistent with the theory depends much on the viewpoint. A particular fact can contradict one theory, and at the same time can be consistent with another theory. Contradictions can thus act as a driving force for changing the individual theories. Each competing theory will then follow a certain evolution path. This process continues until, presumably, some theory is obtained that is found to be satisfactory. In BLIP no attention has, however, been paid to the issue of how different parts of theories can be combined. In our view this is important, as it can speed up the process of constructing the required theory.

3.1 Future Directions

Retaining Justifications in the Integrated Theory

We notice that many recent ET tools (such as Aquinas) do not try to represent knowledge in *one* homogenous knowledge base, but rather provide capabilities for representing *different views*. This argument seems to suggest that we *should not* really try to construct an integrated knowledge base. One could argue that integration might not really be desirable.

In our view this argument is false. Knowledge integration *does not exclude* the possibility of *keeping* different views within the system. Keeping this information has the advantage that the system can *explain* its decisions. The system can tell us, for example, where the rule came from (i.e. from which system), and why it was included in the integrated theory. It also possible to let the system revise its theories in the light of new experience or as a result of communication with other systems.

Using Integrated Theory in Further Learning

One of our future aims is to extend the system so as to be able to improve the integrated theory after new information has become available. The system would then function as a `closed loop system´: it would be capable of using the integrated theory as input for further learning. But which part of the theory should we attempt to modify?

One rather obvious strategy tries to focus on the attention on `weak parts´ of the theory so as not to waste too much effort with unnecessary modifications. Identifying weak parts of the theory is a bit like debugging, only it is more complex. In debugging one tries to identify a faulty step, by comparing, for example, the user´s answer with the system´s answer. Here one needs to consider how a particular step contributes to a overall system´s performance. If some particular step degrades the system´s performance, it is a good candidate for further changes.

Murray and Porter (1989) have done some work in this area. They have developed a system called PROTOKI, which is a prototype of a larger system called KI. This system works as a knowledge integration tool. Its aim is to integrate a new piece of knowledge in the existing knowledge base. This process includes the following three main steps.

The first one is called *recognition*. This step is concerned with the identification of the part of existing knowledge that could be affected by new information. The second step is referred to as *elaboration*. In this step, the system tries to determine the consequences of new information and verify whether some anomalies could be detected. An anomaly will arise if conflicting conclusions could be reached using different chains of reasoning, or if the system´s answer conflicts the answer supplied by the user. The third step is called *adaptation*. This step is concerned with the problem of resolving anomalies. The prototype system inspects the explanation that lead to the observed anomaly. Heuristics are used to determine what could be considered the "weakest premise" of the chain of reasoning, and this premise is then modified.

The system described uses a great deal of domain dependent knowledge in the process. There is no harm in using such knowledge - as long as the general principles are clearly spelled out. These have not been described in great depth. So far, PROTOKI has been oriented towards one example. Also, it is not clear what should happen if the user did not quite agree with the changes made by the system.

Overcoming Language Differences Between Agents

Another line that could be investigated is how to overcome the problem of language differences between agents. This problem typically arises when knowledge is elicited from several (human) experts. Experts can disagree on the vocabulary and the meaning. As Shaw and Gaines (1989) have pointed out the same term can have *different* meanings for different systems. They call this situation a *conflict*. *Different terms* may, however, have *identical* meanings. Shaw and Gaines call this situation *correspondence*.

It is easy to see that INTEG.3 deals with the problem of *conflict*, but not of *correspondence*. That is, the system uses empirical evidence to decide whose definition of the given concept is `best´. There is no contention as to the concept names. These are assumed to behave like *rigid designators* (Genesereth & Nilsson, 1987 (p. 234)). Work is under way to construct a system capable of overcoming the problems of correspondence. The prototype system that is described in (Bradzil et al., 1990) can learn to define the concept of one agent using agent´s terminology.

Some of these issues may be rather difficult to resolve. Different agents may not only use different predicate vocabulary in their conceptualization, but also rather *different conceptualizations* (Morik, 1989). The AI community has been concerned with these problems, but no simple solution is in view. Obviously, we cannot provide a simple solution here, although we intend to work on some of these issues in future.

Conclusion

In this paper we have discussed the method of knowledge integration and described the implemented system INTEG.3. Our experimental results have exceeded our expectations. The integrated theory had a significantly better performance than the original theories. This work suggests that the method could be incorporated into some of the existing knowledge acquisition tools. This would provide a way for integrating the knowledge obtained from experts with the knowledge obtained inductively on the basis of data.

Acknowledgements

The authors wish to thank Katarina Morik for some useful comments during a seminar on this topic. These comments made us spell out the differences between knowledge integration and incremental learning with greater clarity. We would also like to thank various anonymous referees for their comments and suggestions. Finally, thanks also to Jeff Schlimmer, Ming Tan, Tom Mitchell, and others at Carnegie Mellon who discussed this work with me during a recent visit.

This work was partly supported by *Project Ecoles* (Esprit 2 Project 3059). The authors wish to express their gratitude to the Commission of European Communities for this support.

References

Bond A,H, and Les Gasser (1988): *Distributed Artificial Intelligence*, Morgan Kaufmann.

Cestnik, B., Kononenko, I., and Bratko, I. (1987): "ASSISTANT 86 : A Knowledge- Elicitation Tool for Sophisticated Users", in *Progress in Machine Learning,* I. Bratko and N. Lavrac (eds.), Sigma Press, Wilmslow, England.

Boose John, Bradshaw J., Kitto C. and Shema D. (1989): From ETS to Acquinas: Six Years of Knowledge Acquisition Tool Development, in *Proc of Third European Workshop on Knowledge-Based Systems*, J. Boose, B. Gaines, and J.G.Ganascia (eds.), Paris, July 1989.

Brazdil, P. and Clark, P. (1990): "Learning from Imperfect Data", in *Machine Learning, Meta-Reasoning and Logics*, P. Bradzil and K. Konolige (eds.) Kluwer Academic Publishers.

Brazdil, P., Muggleton S. (1990): "Learning in Multi-agent Environment", in *Proc. of the Second Workshop on Algorithmic Learning Theory*, S. Arikawa, A. Marouka, and T. Sato, OHMSHA Ltd., Tokyo, October 1991.

Clark, P. and Niblett, T. (1987): "Induction in Noisy Domains", in *Progress in Machine Learning*, I. Bratko and N. Lavrac (eds.), Sigma Press, Wimslow, England.

Clark, P. and Niblett, T. (1989): "The CN2 Induction Algorithm", *Machine Learning 3,* Kluwer Academic Publishers.

Emde W. (1987): "Non-cumulative Learning in METAXA-3", in *Proc. of IJCAI-87,* Mlian, Italy, pp. 208-212.

Emde W. (1989): "An Inference Engine for Representing Multiple Theories", in *Knowledge Representation and Organization in Machine Learning*, K. Morik (ed.), pp. 148-176, Springer Verlag.

Gaines B. (1989): Knowledge Acquisition: the Cintinuum Linking Machine Learning and Expertise Transfer, in *Proc of Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, J. Boose, B. Gaines, and J.G. Ganascia (eds.), Paris, July 1989.

Gams M. (1989): "New Measurements Highlight the Importance of Redundant Knowledge", in *Proceedings of 4th European Working Session on Learning*, K. Morik (ed.), pp. 71-80, Pitman - Morgan Kaufmann.

Genesereth, M. and Nilsson, N. (1987): *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann Publishers Inc.

Janikow C.Z. (1989): "The AQ16 Inductive Learning Program: Some Experimental Results with AQ16 and Other Symbolic and Nonsymbolic Programs", Rep. of AI Center, George Mason University.

Morik K. (1989): "Sloppy Modelling", in *Knowledge Representation and Organization in Machine Learning*, K. Morik (ed.) pp. 107-134, Springer Verlag.

Murray K.S and Porter B.W. (1989): "Controlling Search for the Consequences of New Information During Knowledge Integration", in *Proc. of 6th International Workshop on Machine Learning*, A.M. Segre (ed.), Ithaca, New York, Morgan Kaufmann Inc.

Quinlan J.R., (1986): "The Effect of Noise in Concept Learning", *Machine Learning (vol. II)*, R. Michalski, J. Carbonell and T. Mitchell (eds.), Morgan Kaufmann Publishers, Inc.

Rada R. and Mili H. (1989): "A Knowledge Intensive Learning System for Document Retrieval", in *Knowledge Representation and Organization in Machine Learning*, K. Morik (ed.), pp. 65-88, Springer Verlag.

Shaw M. & Gaines B. (1989): Knowledge Acquisition: Some Foundations, Manual Methods and Future Trends, in *Proc. of Third European Workshop on Knowledege Acquisition for Knowledge-Based Systems*, J.Boose, B. Gaines and J.G. Ganascia (eds.), Paris, July 1989.

Steels L. and van de Velde W. (1989): "Learning in Second Generation Expert Systems", in *Machine and Human Learning*, Y. Kodratoff and A. Hutchinson (eds.), Michael Horwood Ltd.

Torgo L. (1990): "Incremental Rule Learning Using Entropy Based Search"

Utgoff P.E. (1988): "ID5: An Incremental ID3", in *Proc. of 5th International Workshop on Machine Learning*, J. Laird (ed.), Ann Habour, Morgan Kaufmann Inc.

Van de Welde W. (1989): "Re(presentation) Issues in Second Generation Expert Systems", in *Knowledge Representation and Organization in Machine Learning*, K. Morik (ed.), pp. 107-134, Springer Verlag.

Wielinga B.J. & Breuker J.A. (1986): Models of Expertise, in Proc. of ECAI-86, L, Steels (ed.), Brighton.

Zhang J. and Michalski R. (1989): "Rule Optimization via SG-TRUNC Method", in *Proceedings of 4th European Working Session on Learning*, K. Morik (ed.), pp. 251-262, Pitman - Morgan Kaufmann.