IJCAI-99, Workshop ML-5: Automating the Construction of Case-Based Reasoners. Stockholm 1999. S. S. Anand, A. Aamodt, D.W. Aha (eds.). pp. 39-43.

Knowledge Elaboration for Improved CBR

Frode Sørmo   Agnar Aamodt

Department of Computer and Information Science, NTNU
N-7491 Trondheim
Norway

Abstract

When using knowledge-intensive case-based reasoners, a key issue is to keep the knowledge-base updated. This is usually done by having the user update the knowledge during the retain phase, but machine-learning methods can also be used. In this paper we discuss how an elaboration phase, in which plausible inference rules are derived, can be done in a goal-driven manner to increase the system's automatic learning capability.

1 Introduction

When faced with a new problem case, some case-based reasoners use a knowledge base containing a model of the domain in question. This knowledge base is used to measure the similarity between the problem case and the cases already solved and stored by the system. Although this approach, known as knowledge-intensive case-based reasoning, has shown promise in open and weak domains, it does not in it self solve the problem of indexing and measuring similarity between cases. The quality of indexing and similarity measure in knowledge-intensive methods rests on the shoulders of the knowledge base. If the domain knowledge is insufficient, faulty, or otherwise unable to distinguish between important and spurious aspects of problem areas, then the domain model will not perform well as a basis for similarity measure. The reasoning process depends on a knowledge base that is up to date and contains knowledge organized and kept in a manner relevant to the reasoning processes.

In some domains it is sufficient to build a knowledge base in advance through normal knowledge acquisition techniques. The knowledge-base can then be tested and tuned to the reasoning process in advance. In other domains, especially weak-theory domains, the knowledge base needs to be frequently updated, or gradually refined, in order to be able to relate concepts introduced by new problem cases. The trend in knowledge-intensive systems to date has been to acquire the knowledge required during the retrieval process in a dialogue with an expert user who is presumably able to answer questions the system may have. In the PROTOS system [Porter et. al. 1990], this is the chief method used both to build and update the knowledge base, and in the CREEK system [Aamodt 1991], this method is used to update the pre-designed knowledge-base. While the goal of these systems is to gain autonomy from the user by being able to answer questions by consulting its own knowledge base, the system is in many cases constrained by time-limits for when a confirmed solution to the current problem case is expected. This makes it harder to employ computer-intensive machine-learning techniques such as data-mining to find the desired knowledge. When neither the system nor the person using the system at the time is able to answer a question, the problem-solving process may have to be suspended. This is especially the case with questions related to earlier cases the current user may not be familiar with. By reflecting on the sufficiency of knowledge "surrounding" a case to support retrieval in the future, knowledge acquisition goals may be identified at an earlier time. During a phase of elaboration beginning when a case is retained, other methods may be employed than during the retrieval. This paper will suggest two such methods for updating the domain model, and look briefly at how they can be used in a goal-driven manner. The methods discussed are data-mining on related databases, and searching for new plausible inference rules by analyzing the existing knowledge. In CREEK this knowledge has the form of a multi-relational, dense semantic network in which each node and each link are concepts represented as frames.

2 The Phase of Elaboration

The time from a case is retained until it is used to solve an actual problem, may be quite a while longer than the time available when trying to solve an actual problem. During this time, computer-intensive methods can be used to elaborate on the knowledge contained in the system. Currently, we are in our group studying two such methods. One is data-mining, where databases external to the case-based reasoner may be used to find new knowledge for inclusion in the knowledge-base of the case-based reasoning system. The other method is to find plausible, domain-specific inference rules. In particular, we are looking for plausible inheritance rules, which enable new inferences to be drawn that will extend the explanatory capability of the general domain model. Once a suitable set of such rules are found, they will improve the utilization of the general background knowledge within the system's retrieval, reuse, and case retaining phases. We are also interested in studying how methods such as these two may work together, for instance by studying to which degree data mining methods can contribute to the generation of plausible inference rules in the general domain model. The topic of this paper, however, is to identify the role and structure of plausible inheritance rules, related to improving case-based reasoners by automatic means. The details of the methods to achieve that are currently under development.

2.1 Data-mining

Since the aim of the data mining methods we study is to improve the domain knowledge that support the CBR processes, Bayesian network has been identified as a promising approach [Aamodt and Langseth 1998]. A Bayesian network (BN, also referred to as belief networks) is a network of statistical dependencies between parameters. They may model a network of causal relationships, of other types of influences between concepts. BNs are constructed partly by manually defining the important concepts/parameters and their dependencies, partly by automatic methods that update the strengths of the dependencies.

By using a large database of parameters and values, a BN may be generated – or it may update an existing domain knowledge model. Hence, new relationships between domain concepts may be discovered, and dependencies already existing within the domain model may be updated or confirmed. The result is a submodel of statistical relationships, which will live their own life in parallel within the more general domain knowledge model. The BN generated submodel is dynamic in nature i.e. the strengths of the dependencies will continuously be updated as new data is seen. This is opposed to the static relationships of a classic domain model, and is potentially useful for supporting the acceptance or rejection of hypothesized plausible inheritance rules.

2.2 Plausible inference rules

Property inheritance over links describing taxonomic hierarchies (ISA-relations) is quite common in knowledge representation languages. The common property inheritance can be visualized as in figure 1.

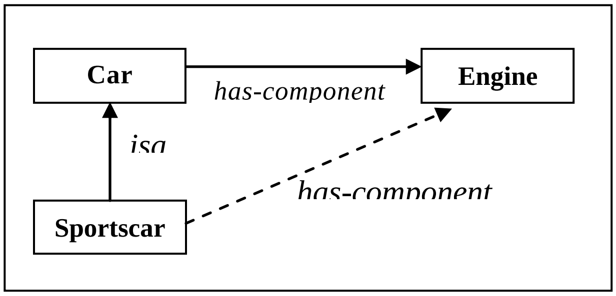


Figure 1

The general inference rule of this inheritance can be written as in rule 1 below.

For any relation R,

if **x** *ISA* **y** and
y R **z**

---

then **x** R **z**

Rule 1

Although property inheritance over ISA-links is quite common, other relations may also allow properties to be inherited over them, depending on the domain. In a paper on plausible inference, Cohen and Loiselle [1988] use the example of their favorite cough syrup, which happens to contain alcohol. In this example, the cough syrup should inherit the intoxicating property of the alcohol through a *containes* relation, as shown in figure 2.

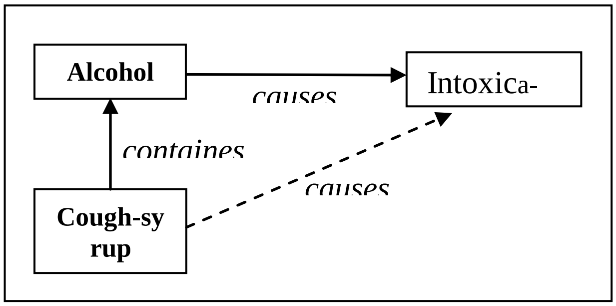


Figure 2

In the case of cough-syrups, and indeed the whole domain of medical liquids, the rule that the properties of one component is transferred to the composite product may be a good general rule. The rule may be expressed as in rule 2.

if **x** *CONTAINS* **y** and
y *CAUSES* **z**

then **x** *CAUSES* **z**

Rule 2

By identifying inference rules as the one above, the knowledge base would be able to gain more autonomy from the user as knowledge can be inferred from the current knowledge base to a larger extent than without it. The challenge is in finding exactly which of these rules are plausible. Cohen and Loiselle points out that a knowledge base of N relations will produce (N2 * N) / 2 pairs of relations and thus the same number of possible rules. In their experiment, examples were built from the rules, using actual concepts covered by the rule. The instantiated rules were then presented to human testers who would judge their plausibility. Cohen and Loiselle found that certain structural characteristics of the rules predicted well if they were found to be plausible. The best predictor of a plausible rule, was found to be the rule's transitivity. A transitive rule can be described as in rule 3.

For any relation R,

if **x** R **y** and
y R **z**

then **x** R **z**

Rule 3

An example of an instantiated transitive rule is found in figure 3.

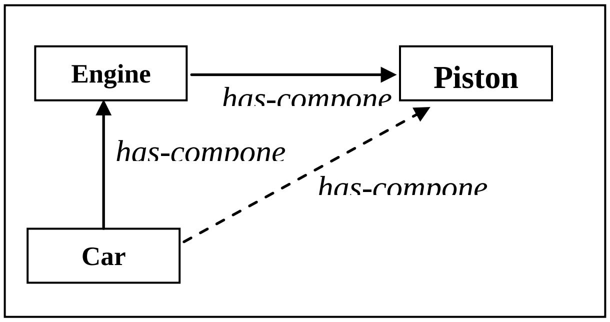


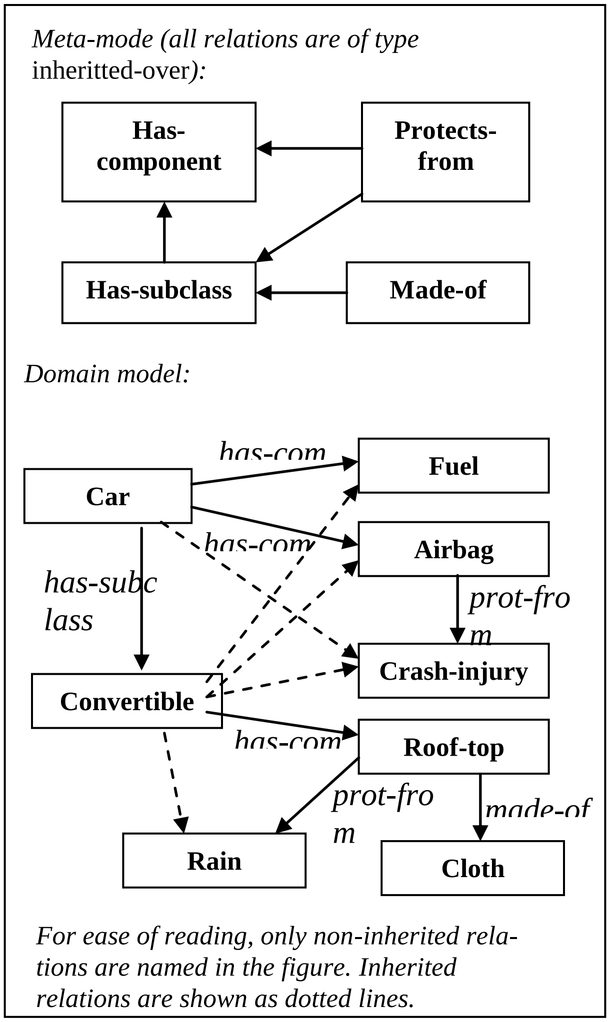
Figure 3

Other structural characteristics was also found, but while these experiments showed that such structural characteristics can be good predictors, they also made it clear that the rules need to be verified.

Within the framework of case-based reasoning our hypothesis is that it is possible to increase the accuracy of the predictions of which inference rules are plausible by checking rule hypothesis against the case-base. Cases containing both ends of a chain of CAUSES-relations could, for instance, support an inference rule suggesting that the causal relation is transitive. This is especially interesting for more domain-specific rules. While it may be true that when mixing food, drink or couch-syrup, the properties of the individual ingredient in many cases are retained in the composite product, the same may not be true in other domains. A car containing fuel, for instance, does not inherit the explosive properties of fuel, except maybe in the movies.

When a set of rules are found, we can use these rules to improve the standard inheritance mechanism used in most semantic nets. In our ongoing implementation we use Creek's frame-based representation, where each relation-type is represented as an entity in a meta-knowledge model. As new rules of inheritance are found, new relations between the entities describing relation-types are added. If it is found that the *causes* relation always seems to be able to transfer over the *contains* relation in this domain (as in figure 2 and rule 2) , then a new relation between the entity describing the *contains* relations and the *causes* relation will be added to signify that the *causes*-relation may be inherited over the *contains*-relation. This information, then, will be used by the inheritance mechanism of Creek. While the standard inheritance mechanism only searches for inherited relations over *subclass-of* relations, our new inheritance mechanism searches all relations, but only inherits relations that are listed as possible to inherit in the relation-types entity in the meta-model. For a visual example, see figure 4. In that example, the *convertible*-entity is shown with relations and inherited relations. As the figure shows, the *convertible* not only inherits the *has-component*-relations from the *car*-entity over the *has-subclass* relation, but also the *protection-from* relation from *rooftop*-entity over the *has-component* relation. Also, as the *car*-entity inherits the *protects-from* relation from *airbag* to *crash-injury*, the *convertible*-entity also inherits this relation over the *has-subclass*-relation from the *car* to the *convertible*.

With this new inheritance method, the inheritance mechanism may have been taken a step towards a more general rule-based mechanism. We still call it inheritance, as it is made for a semantic-net based knowledge representation, and also because it only can deal with rules that specify what relations can be transferred (or inherited) over some type of relation. Still, this extended inheritance certainly will be somewhat slower than the standard inheritance mechanism, and we may have to wrestle with some of the problems of more general rule-based methods, such as assuring that the method will not enter a loop.



For ease of reading, only non-inherited relations are named in the figure. Inherited relations are shown as dotted lines.

Figure 4

3 Goal-driven elaboration

While the methods described above may individually find useful information, we hope that by using both methods within the same framework, we can gain accuracy by comparing the results. As previously mentioned, data residing in ‘ordinary’ data bases contain dependencies that can be revealed through, e.g., Bayesian Network methods, and hence provide an additional source for justification of possible plausible inheritance relationships. Also, we hope that such a framework would be able to give information to the machine-learning methods about what knowledge are sought. This is important because even when the computer-intensive methods find knowledge between problem-solving sessions and increase the knowledge in the case-based system, random pieces of knowledge do not necessarily increase the quality of the knowledge-base. Learning through methods effectively searching for pieces of knowledge in the dark may add to the knowledge without adding to the quality and efficiency of the reasoning. This is one of the reasons why generalizing after solving a problem goes contradictory to the case-based reasoning core idea that no inductive inferences should be done until the problem-case is known. When the problem is known, it is possible to use the problem context and specific information as a bias for the induction – information not available until the system has an actual problem case. However, recognizing that the knowledge base is lacking in knowledge to support the reasoning process can be done in other phases of the case-based reasoning cycle, as the methods using the knowledge is known in advance. Analyzing the knowledge needs of the algorithms using the knowledge base to, for instance, generate explanations between differences in cases with the same solution, may result in the system discovering weak points in the knowledge base. This process can be as simple as trying to retrieve the case just solved by using its problem description – a method already common in case-based reasoners to check if the case is well indexed. More advanced methods could include:

- Creating plausible new problem-descriptors by using properties found in other cases with the same solution.
- Checking for inconsistencies between the just-solved case and the relevant parts of the domain model.
- Storing exceptions to general rules in the domain model and taking action when many exceptions suggest a lack of knowledge (as done in the dynamic memory model of Schank [1982]).
- Compare the just-solved case with other cases with the same solution to find possible missing relations, for instance by a process of structure mapping and comparing the structure of the cases [Gentner 1987].

When learning goals are identified, there is still the challenge of finding ways to goal-direct the knowledge acquisition methods. The methods described in this paper are not often thought of as particularly goal-driven. The usefulness of data-mining, in particular, lays in its ability to discover knowledge no one knew existed, and even knowledge no one knew would be useful. Still, data-mining methods tend to give information judged irrelevant by humans in addition to the relevant knowledge. It may be that learning goals can serve as relevance filter after or during mining. While such a data-mining method is not currently in development, our ongoing implementation of the search for plausible inference rules will be able to focus on the goals given to it by the case-based reasoner.

4 Conclusion

This paper describes work very much in-progress, and as such we are unfortunately not able to give good examples and results at the time of writing. We have, for instance, no evidence that the elaborated knowledge will increase the accuracy of indexing and selection of cases, but our hypothesis is that this will be the case. To illustrate how this may be, we can go back to the example of the couch-syrup and imagine a scenario where we want to know why a person is intoxicated. To find these cases, it would be quite beneficial to the system to know that any liquid containing another may inherit the properties of the contained liquid. If this is known, we will be able to retrieve not only cases linked directly to alcohol, but also to cases linked to liquids containing the intoxicant – such as couch-syrup. We will also be able to explain the difference in problem-case and the cases found. If our solution case is a person suffering from intoxication after drinking too much of another alcoholic liquid, the system can explain that difference by pointing out that both liquids contain alcohol.

The goal of this research is to find this type of information in other than asking the user during the retrieval phase. This is important, as the user may not be familiar with cases asked about, and he or she may not be able to answer the question. If such questions are identified earlier, they may be sent to experts, or being subjected to computer-intensive machine-learning methods. We have suggested two such methods in this paper: searching for plausible inference rules, and the use of data-mining on databases outside the case-based reasoner. Although these methods can find knowledge without having learning goals, we are also investigating how they perform in a goal-driven framework similar to the one introduced by Ram and Leake the book “Goal Driven Learning” [1995, chapter 1].

References

[Aamodt 1991] Agnar Aamodt, *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. Ph.D. Dissertation, University of Trondheim, Department of Electrical Engineering and Computer Science.

[Aamodt and Langseth 1998] Agnar Aamodt, Helge Langseth. Integrating Bayesian Networks into Knowledge-Intensive CBR. Proceedings of AAAI Workshop on CBR Integration, Madison July 1998. Technical Report WS-98-15. AAAI Press.

[Cohen and Loiselle 1988] Paul R. Cohen and Cynthia L. Loiselle. Beyond ISA: Structures for Plausible Inference in Semantic Networks. *Proceedings of the 7th National Conference on Artificial Intelligence*. pages 415-420, St. Paul, MN, August 21-26, 1988. American Association for Artificial Intelligence.

[Gentner 1987] D. Gentner. The Mechanisms of Analogical Learning, *Similarity, Analogy, and Thought,* editors S. Vosniadou and A. Ortony. Cambridge University Press, New York, 1987

[Porter et. al. 1990] Bruce Porter, Ray Bareiss, Robert Holte. Concept learning and heuristic classification in weak theory domains. *Artificial Intelligence* 45 (1-2), Sept. 1990, pp. 229-263.

[Ram and Leake 1995] Ashwin Ram and David B Leake, editors. *Goal-Driven Learning.* MIT Press, Cambridge, Massachusetts, 1995.

[Schank 1982] Roger Schank, *Dynamic memory; a theory of reminding and learning in computers and people.* Cambridge University Press. 1982.