Proceedings of the Sixtennth Annual Conference of the Cognitive Science Society, 1994

Using Introspective Reasoning to Guide Index Refinement in Case-Based Reasoning

Susan Fox and **David Leake**
Computer Science Department
Lindley Hall 215
Indiana University
Bloomington, IN 47405
E-mail: sfox@cs.indiana.edu, leake@cs.indiana.edu

Abstract

Case-based reasoning research on indexing and retrieval focuses primarily on developing specific retrieval criteria, rather than on developing mechanisms by which such criteria can be learned as needed. This paper presents a framework for learning to refine indexing criteria by introspective reasoning. In our approach, a self-model of desired system performance is used to determine when and how to refine retrieval criteria. We describe the advantages of this approach for focusing learning on useful information even in the absence of explicit processing failures, and support its benefits with experimental results on how an implementation of the model affects performance of a case-based planning system.

Introduction

Case-based reasoning (CBR) has been widely investigated both for its practical applications and as a model of human reasoning and learning (see Kolodner (1993) for an overview). One relevant facet of human reasoning, however, has received little attention in case-based reasoning research: the role of metacognitive learning. Although people can change their metacognitive strategies (e.g. Flavell (1985)), CBR systems generally rely on fixed reasoning strategies determined in advance by the system designer.

We are investigating how introspective reasoning and learning can be applied by case-based reasoning systems to improve their processing strategies. Although learning is fundamental to case-based reasoning, current CBR systems learn primarily domain knowledge, in the form of new cases that are added to their case libraries. They do not refine the strategies that they use to retrieve and apply their cases. Introspective learning in CBR systems is needed to enable them to analyze their reasoning performance and learn to apply their domain knowledge more effectively.

This paper describes a general framework for introspectively applying model-based reasoning techniques (see Davis (1988) for an overview) to case-based reasoning. It focuses on its application to refining indexing knowledge by retrospectively analyzing and suggesting revisions to the case retrieval process of a CBR system. The basic premise of the model-based approach is that if a CBR system has access to information from a *model* of its ideal behavior, it can note deviations and respond by learning to better approximate the ideal behavior characterized by the model. The approach used here was proposed by Birnbaum et al. (Birnbaum, Collins, Brand, Freed, Krulwich, & Pryor, 1991), who have applied model-based methods to the generation of self-debugging planners in other contexts (Collins, Birnbaum, Krulwich, & Freed, 1993; Birnbaum, Collins, Freed, & Krulwich, 1990). A number of issues arise in applying this approach, including how to represent the model, the levels of abstraction at which the model should characterize system behavior, and how to apply the model to detecting and repairing problems.

In this paper, we focus primarily on one aspect of the research on model-based introspective learning in CBR: its role in index refinement. In particular, we show that introspective model-based reasoning for index refinement can detect problems that cannot be detected by previous approaches to introspective failure-driven learning. We begin by discussing the use of model-based reasoning for introspective learning and the benefits of the approach for (1) extending the class of processing problems that can be detected and (2) guiding learning in response to those problems. We then describe the process by which our computer model refines the indices used to store cases, and evaluate the model's performance on learning a new class of indices by introspective reasoning. We close by discussing the issues and directions that have arisen from this research.

Introspective Model-Based Reasoning for CBR

The fundamental principle of model-based reasoning is to use a model of correct device behavior to identify and diagnose failures of the device. That model can be expressed in the form of assertions that are true when the model is functioning correctly. Although model-based reasoning is usually used to diagnose and repair the failures of a device to perform as expected, it has been observed (Collins et al., 1993; Birnbaum et al., 1990) that model-based reasoning can also be used to characterize *ideal* system behavior—beyond expectations for actual performance—in order to identify deficiencies and guide learning to bring the system closer to the ideal.

Birnbaum et al. sketch this approach with a hypothetical example involving the assertion that in an ideal CBR system, *the case retrieved for any problem is the case in memory closest to the final solution*. (With appropriate similarity metric, the "closest" case will be the case that is easiest to adapt.) If the assertion does not hold, the CBR system may still generate good results, but only by expending adaptation effort that would have been unnecessary if a more appropriate case had been retrieved. The problem of refining indexing in response to the failure of this assertion will be the central focus of the rest of this paper.

We integrate model-based reasoning with a case-based reasoning system by having the model-based reasoner monitor

the reasoning processes of the CBR system as it performs, comparing actual performance to the ideal model. When a discrepancy is discovered, the model is used to perform credit assignment and to suggest alterations to the reasoning processes to improve their performance.

Motivations

The importance of index refinement

Although case retrieval in current CBR systems is often based on criteria pre-selected by the system designer, modeling index refinement is important for both practical and cognitive reasons. A pre-defined set of indices is unlikely to be adequate for real-world tasks: poorly-understood task domains and changing circumstances in real-world situations may make it may impossible to predict *a priori* which indices should be used to store and retrieve cases. Likewise, modeling the refinement of indexing criteria is important for CBR as a cognitive model. For example, psychological experiments have demonstrated a developmental shift in human criteria for determining important features in analogies (Gentner, 1988); children of different ages use different relevance criteria when determining the features to map when applying a story to new circumstances, with older children using deeper features. A central question is how this shift can occur.

Expanding failure-driven methods for index refinement

Approaches to index refinement in previous CBR systems are generally based on explicit processing failures such as the retrieval of a case that turns out unexpectedly not to apply to a new situation (e.g., Hammond (1989)). In general, CBR systems learn when they reach an impasse during processing, when the results of a processing step conflict with expectations, or when the solution produced by its reasoning proves unsuccessful. These are examples of impasse-driven or failure-driven learning, a learning paradigm that has been extensively studied in many contexts. Such methods are a valuable tool for improving system performance. However, when the efficiency of the reasoner's processing is a concern, inadequate behavior may not always be revealed by impasses or outcome failures.

Identifying hidden failures by model-based reasoning

When the result of processing is successful and no obvious impasses have been found, failure-driven learning systems assume that there is nothing to be learned from the processing episode. However, learning may be necessary despite the successful outcome of processing. For example, case-based reasoning systems can be thought of as having three components: a case retriever, an adaptation component to fit the retrieved case to the new situation, and an evaluation component to assess the quality of the resulting solution. In principle, a sufficiently powerful adaptation component could compensate for retrieval failures by adapting even an inappropriate prior case into a correct solution (e.g., by deleting all components of the retrieved case and reasoning from scratch). However, if a relevant case exists in memory and can be applied with little adaptation, it should be less expensive to start from that case than to reason from scratch.

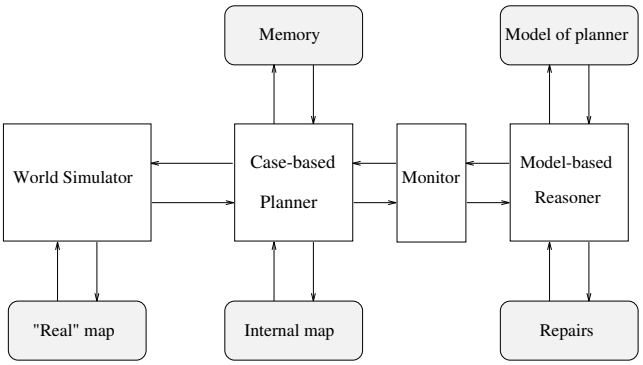

Figure 1: Architecture of Robbie

Previous research has addressed the question of how to store and replay the reasoning from prior solution efforts (e.g., Carbonell (1986)), to avoid having to re-generate that reasoning from scratch and to avoid prior explicit failures during processing. However, if deficient processing results in a successful result, learning to replay prior reasoning will simply cause the deficient processing to be replayed in the future. We propose augmenting impasse- and outcome-based learning with model-based analysis of the solution process itself. during processing. Such analysis will permit changing the underlying processing strategies to improve sub-optimal processes even if they result in successful outcomes.

A difficult issue raised by this method is the credit assignment problem: how to identify the cause of the failure. When a system attempts to learn from hidden problems that do not cause outcome failures, as we propose, the credit assignment problem is compounded with a serious *failure identification problem*. Before attempting to ascribe blame for a processing problem, the reasoner must identify that problem. Traditional methods for detecting failure, such as comparing results of processing steps to expectations (e.g., Hammond (1989)), do not enable problem detection when the outcomes of the steps are sufficient. Our approach to identifying poor performance despite a successful final solution is to use hindsight: information provided by the successful solution is used to illuminate how the solution should have been generated.

In particular, when a CBR system attempts to retrieve a case to apply to a new situation, case retrieval must be based on partial knowledge of the relevant features of the problem—the features that are relevant to retrieval may not be apparent until the problem has been solved. The need to base retrieval on partial knowledge has been addressed in CBR research through methods such as *situation assessment* (e.g., Kolodner (1993, pp. 371-372)), which attempts to reason about important retrieval considerations prior to formulating a query to memory. However, that process necessarily uses partial information: the information available before the solution has been generated.

Once a solution has been generated, it can play a valuable role in retrospectively analyzing the effectiveness of indexing and in refining the indexing process. For example, a visitor to a city with only a few bridges over a dividing river may not initially realize the importance of taking the location of those bridges into account when planning to get from one place to another. If the visitor tries to adapt a plan that starts from a

- Turn north on the corner of Maple and Elm
- Move north along Elm to the north side of Birch
- Turn east on the north side of Birch
- Move east to the position 20 feet along Birch

Figure 2: High-level plan

nearby point on the other side of the river, she may be blocked by the river and forced to backtrack to find a bridge to cross. Initially, she will have no way to anticipate the problem, but after reaching the destination it will be clear that a better plan would have been to aim towards the bridge—not the final destination—first.

In our approach, the quality of initial case retrieval is judged by examining the solution finally generated and then using information from that solution to retrieve the case with the most similar solution. Before the problem has been solved, retrieval must be based only on the system's goals, because the solution is unknown; after problem solution, both goal-based and solution-based features are available. The case with the closest solution is compared to the case that was retrieved originally and to the final solution to identify the features that should have been given more emphasis in the original retrieval process. That analysis is used to refine indexing criteria.

The Computer Model

The theory is being investigated in context of case-based path planning. The implemented system, Robbie[1], integrates an introspective reasoner with a case-based planning system, as illustrated in Figure 1. The case-based planner interacts with a simulated world, and passes information through a monitoring system to the model-based reasoner. The model-based reasoner examines the reasoning processes of the planner during the planning process, and compares the reasoning to its assertions about ideal behavior. It suggests repairs for deviations from its ideal, whether the deviation was an explicit failure or merely sub-optimal behavior.

Given a desired destination as input, the system generates an initial skeletal plan by case-based reasoning: retrieving a prior plan and adapting it to create high-level plans such as the example in Figure 2. That information is provided to a component that uses the skeletal plans as an outline for guiding reactive execution of the plan by a simulated robot pedestrian, using a Reactive Action Package model of plan execution and monitoring (Firby, 1989). Because execution/evaluation of the plan is done reactively, the final solution may be optimized during execution. Although this project investigates issues in plan execution and integrating high-level planning with a reactive component, those issues are beyond the scope of this paper.

Initial plan retrieval criteria

The planning system begins with an extremely simple retrieval criterion: Given a new destination, it attempts to retrieve previous plans with the most similar starting and ending locations. Although this seems a reasonable heuristic, it is in fact

[1]Re-Organization of Beliefs By Introspective Evaluation

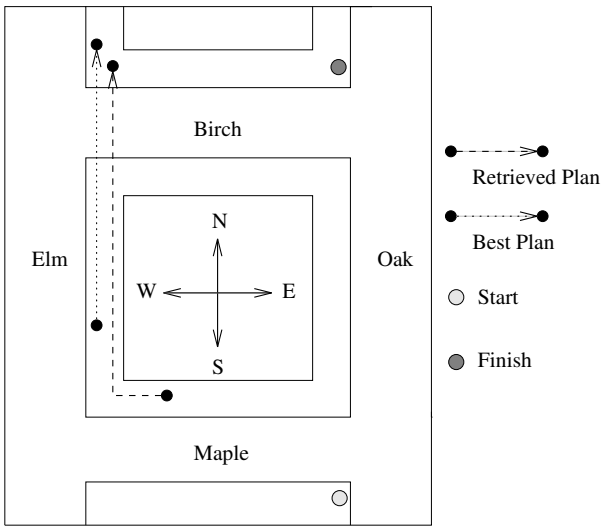

Figure 3: Example of a Failure

inadequate. Using only endpoints ignores special features of the path between them, including the relative positions of the endpoints and obstacles that may lie in the way. An example of failure to attend to path features is the retrieval of a plan appropriate for travel from a nearby spot across a river, as discussed previously.

Problems that may arise are illustrated in the following implemented example. The robot pedestrian starts at the indicated location on the map shown in Figure 3, on the sidewalk at the corner of Maple and Oak Streets, with the goal to reach the northwest corner of Oak and Birch Streets. As is clear from the map, the optimal plan would be to move directly north along Oak street. There is a plan available that involves this *movement*, but because of the initial retrieval criteria—only the similarity of origin and destination locations—the planner does not retrieve it. Instead, the planner retrieves a plan that involves starting on Maple and ending on Birch, just as the goal does, whereas the real "closest" plan, judged in terms of the adaptation effort required, does not involve Maple at all. Because the retrieval criteria do not recognize the importance of plans which move in one direction versus plans which move in more than one, the planner must expend extra effort adapting a suboptimal plan.

The adapted plan will include additional steps for moving to the corner. The reactive execution component will discard those steps as already completed, optimizing the final solution to a simpler form.

The introspective reasoner monitors the planner's reasoning throughout, and is signaled whenever the planner is available to be verified. Up to the point when execution finishes successfully, no assertions have failed. At this point, the reasoner is signaled again. The reasoner then checks the currently relevant assertions, such as *the case retrieved is the closest one to the final solution*. In comparing the final solution (a straight line between the starting and ending locations) to the plans in memory, it is obvious that the retrieved plan does not have the most similar solution.

Table 1: Effects of introspective index learning

Condition	Case library size	% retrieved	% successful solutions
Selected test cases:			
Case learning alone	3-7	75	75
Case and index learning	3-9	45	100
Random test cases:			
Case learning alone	3-21	66	47
Case and index learning	3-26	34	60

The learning process

Having found one failure, the reasoner performs credit assignment by considering other assertions of its model causally related to the current failure. The initial failure discovered may not be the root cause of the problem; other causally related failures will help to determine which aspects of the system architecture are likely to be responsible for the problem. The assertions in the model of the CBR process are organized by the parts of the system to which they refer, and contain causal links to other relevant assertions in the model. Starting from the initial failed assertion other assertions which are generalizations or specifications of it will be considered, as well as those in other parts of the model which are causally related.

For example, failing to retrieve the closest case could be caused by a failure in judging cases during retrieval, or by a failure in constructing the index used to retrieve. It could not be caused by faulty adaptation, so assertions about adaptation will be ignored. The system considers assertions related to failures in retrieval, judgements of similarity in retrieval, and the choice of features used in index construction.

In this case it determines that the planner failed to recognize a shared feature of the closest plan and new situation: that both involve moving in one direction with no turns. Because the lack of that feature led to a faulty retrieval, it contradicts one of the model's assertions (that the constructed index will include all relevant features). The system repairs the problem by creating a new rule for constructing indices that identifies the shared feature (moving in one directions north/south). The system uses the rule to re-index the cases in memory and stores it for future use.

The Value of Introspective Learning

Although the model of the introspective reasoner is still very incomplete, we have begun assessing the effects of introspective learning in small case libraries. This initial evaluation of indexing performance is promising. We performed two sets of tests, the first using a selection of eight goals selected for their relative similarity to members of an initial three-case library (such goals should be easy for the system to fulfill). The system was tested on these goals with (1) both learning of cases and introspective refinement of indexing, and (2) learning of cases only. Because the planner chooses randomly between retrieved cases that are judged equally similar, all three sets were run five times and results were averaged to cancel out random fluctuations.

Recall that the system starts with minimally restrictive retrieval criteria, involving only the superficial features of the starting and ending locations. It retrieves all cases considered equally similar and selects randomly among them. Sharper retrieval criteria will focus retrieval on a smaller set of cases, but too restrictive retrieval may overtax a limited adaptation module. Consequently, the performance of the overall system was judged according to two criteria: (1) the number of cases that were considered potentially relevant and retrieved, and (2) the number of successfully adapted and applied cases.

Our results are summarized in the first part of Table 1. Introspective learning did help to focus retrieval: the average number of cases considered compared to the total number of cases in memory was 45% when using introspective index refinement, but 75% when using only case learning. Perhaps more important than the savings in retrieval, introspective index refinement led to the retrieval of better cases. The tests with introspective learning disabled failed to create viable plans for two out of the eight goals. In those cases, the system retrieved inappropriate cases based on the starting and ending locations alone. Previous situations had produced suboptimal plans that happened to succeed; without learning from index refinement those cases were not reindexed, causing a catastrophic failure in the later situations.

There are two concerns for judging results of the first test. One is whether the results will scale up as the case library grows, given the sparseness of the initial library. Another is that the sequence of goals selected skews the results; the goals were tailored to the contents of the initial case base and were therefore easier to match and adapt than randomly chosen locations. To address those questions, our second test compared case learning alone to case and introspective learning while building up a case library of 20-25 cases (from the same initial library), using a randomly-generated sequence of goals. The goals were chosen randomly, but were restricted to problems theoretically within the planner's capabilities. The same random sequence was used for both sets, and each was run five times as above. The results are summarized in the last half of Table 1. Because both the model and planner are still incomplete, even with introspective learning there were some goals for which the system failed to generate a successful plan. Still, the improvement of case and introspective learning over case learning alone is marked, both in terms of percentage of the case base retrieved and percentage of successful solutions.

Issues and directions

Our research to this point as identified three major issues now under investigation:

Table 2: Failure Types

Component	Failures	Assertions
Retriever	Failed to retrieve any case	The retriever will find some match
Adaptor	Did not completely adapt the case	The adaptor has a strategy for every situation
	Made a bad adaptation	The adaptor will improve the case
Executor	Failed to reach goal	The robot will end at the goal
	Failed to complete plan in time	The robot will take N timesteps for each step
Storer	Stored a plan twice	The storer will recognize identical plans

Developing a case-based approach to the CBR process itself

This paper has addressed one facet of using model-based reasoning to refine the reasoning strategies used in case-based reasoning. A key issue is applying these methods to other parts of the processing of CBR systems as well. This effort can be facilitated if each component of the CBR system is developed using similar mechanisms. We have chosen to treat each component of the system itself as a CBR system (Kolodner & Simpson, 1989), allowing the general model of case-based reasoning as a whole to be applied to introspective reasoning and learning about specific sub-components as well. For example, anticipating goal interactions and doing case adaptation are both viewed as case-based processes themselves, and are implemented with the same mechanisms as the planner itself. As this approach is refined, we expect it to provide leverage for extending the model-based reasoning process to perform introspective reasoning about all parts of the CBR system.

Developing and applying a multi-level model

Determining what information to include in the model, and how to structure it, are central issues. In Birnbaum et al.'s proposal (Birnbaum et al., 1991), the model of CBR contains very high-level assertions applicable to the broad class of case-based planners. While such assertions cover a wide range of failures, they are not operational for identifying appropriate repairs for failures. We are developing a hierarchical model spanning the spectrum of generality from highly abstract assertions to assertions specific to our particular implementation. By using a hierarchy of assertions, the general ideas and structure will be applicable to other systems while we retain the ability to detect and repair specific failures of our system.

Developing a taxonomy of failures and repairs

We are developing a vocabulary of failure types to guide our choice of assertions and to find repair points in the system's reasoning strategies. Table 2 shows a sample set of failure types and a few related assertions. For example, the failure type "failing to complete adaptation" leads to investigation of assertions about how to gauge the progress of adaptation. A high-level assertion like "Adaptation will complete" is very general and easy to connect to other parts of the system, but it is not an assertion we can check. More specific assertions like "Adaptation will take no more than X steps" are easy to monitor, but lack generality. If the specific assertion fails, it could be because of a failure of the adaptation itself, or it could be because a bad case was retrieved. The high-level assertion connects at the right level of abstraction to related parts of the retrieval assertions which can then be checked in a top-down manner. It is easier to specify a repair for the more specific assertion; there may be more than one way in which adaptation could fail to complete. Such examples point to the need for system-specific assertions to recognize when adaptation has failed, and the need for high level assertions to provide a causal framework. Our current model incorporates both assertions given above.

Related Work

Collins et al. (1993) use a model-based reasoner like the one used here to introspectively improve a learning chess program, CASTLE. Stroulia & Goel (1994) use a "structure-behavior-function" model of a system's functioning to model and improve a multi-strategy planning system, Router, and a design system, Kritik2. Ram & Cox (1994) propose meta-explanations describing kinds of system failures as a means of doing introspective reasoning for a CBR system. Oehlmann, Edwards, & Sleeman (1994) have also done work in re-indexing cases and metacognitive issues for CBR. Other approaches to memory search and index learning are considered in (Leake, 1994; Redmond, 1992; Sycara & Navinchandra, 1989).

Conclusion

In order for case-based reasoners to improve their ability to apply the experiences in their case libraries, they must be able to refine their reasoning strategies. We have presented an implemented framework for using introspective model-based reasoning to refine indexing strategies. In our framework, index refinement is done by retrospectively comparing actual indexing performance to the performance described by a model of retrieval behavior in an idealized case-based reasoning system. This method provides a valuable tool for augmenting learning methods based on explicit failures. By using the model as a yardstick against which to judge its performance, a case-based reasoning system can detect and repair problems that are not manifested by explicit failures during its processing. We are now investigating issues in extending this approach to other components of a CBR system, in reasoning with a multi-level model of the whole CBR process, and in developing a general theory of the failures and repairs that apply to the reasoning strategies of CBR systems.

Acknowledgements

We would like to express our appreciation for the helpful comments of the anonymous reviewers.

References

Birnbaum, L., Collins, G., Brand, M., Freed, M., Krulwich, B., & Pryor, L. (1991). A model-based approach to the construction of adaptive case-based planning systems. In Bareiss, R. (Ed.), *Proceedings of the Case-Based Reasoning Workshop*, pp. 215–224 San Mateo. DARPA, Morgan Kaufmann, Inc.

Birnbaum, L., Collins, G., Freed, M., & Krulwich, B. (1990). Model-based diagnosis of planning failures. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pp. 318–323 Boston, MA. AAAI.

Carbonell, J. (1986). Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. In Michalski, R., Carbonell, J., & Mitchell, T. (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, pp. 371–392. Morgan Kaufmann Publishers, Inc., Los Altos, CA.

Collins, G., Birnbaum, L., Krulwich, B., & Freed, M. (1993). The role of self-models in learning to plan. In *Foundations of Knowledge Aquisition: Machine Learning*, pp. 83–116. Kluwer Academic Publishers.

Davis, R. (1988). Model-based reasoning: troubleshooting. In Shrobe, H. (Ed.), *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*. Morgan Kaufmann, Palo Alto.

Firby, R. (1989). *Adaptive Execution in Complex Dynamic Worlds*. Ph.D. thesis, Yale University. Computer Science Department Technical Report 672.

Flavell, J. (1985). *Cognitive Development*. Prentice-Hall, Englewood Cliffs, NJ.

Gentner, D. (1988). Metaphor as structure mapping: the relational shift. *Child Development*, *59*, 47–59.

Hammond, K. (1989). *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, San Diego.

Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA.

Kolodner, J. & Simpson, R. (1989). The mediator: analysis of an early case-based program. *Cognitive Science*, *13*(4), 507–549.

Leake, D. (1994). Towards a computer model of memory search strategy learning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* Atlanta, GA.

Oehlmann, R., Edwards, P., & Sleeman, D. (1994). Changing the viewpoint: re-indexing by introspective questioning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* Atlanta, GA.

Ram, A. & Cox, M. T. (1994). Introspective reasoning using meta-explanations for multistrategy learning. In Michalski, R. & Tecuci, G. (Eds.), *Machine Learning: A Multistrategy Approach, Vol. IV*. Morgan Kaufman.

Redmond, M. (1992). *Learning by Observing and Understanding Expert Problem Solving*. Ph.D. thesis, College of Computing, Georgia Institute of Technology. Technical report GIT-CC-92/43.

Stroulia, E. & Goel, A. (1994). Task structures: what to learn?. In *Proceedings of the 1994 Goal-driven Learning Symposium*, pp. 112–121 Stanford, CA. AAAI.

Sycara, K. & Navinchandra, D. (1989). Index transformation and generation for case retrieval. In Hammond, K. (Ed.), *Proceedings of the Case-Based Reasoning Workshop*, pp. 324–328 San Mateo. DARPA, Morgan Kaufmann, Inc.