Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, 1992, pp. 313–318

Constructive Similarity Assessment: Using Stored Cases to Define New Situations

David B. Leake
Department of Computer Science
Indiana University
Bloomington, IN 47405
leake@cs.indiana.edu

Abstract

A fundamental issue in case-based reasoning is similarity assessment: determining similarities and differences between new and retrieved cases. Many methods have been developed for comparing input case descriptions to the cases already in memory. However, the success of such methods depends on the input case description being sufficiently complete to reflect the important features of the new situation, which is not assured. In case-based explanation of anomalous events during story understanding, the anomaly arises *because* the current situation is incompletely understood; consequently, similarity assessment based on matches between known current features and old cases is likely to fail because of gaps in the current case's description.

Our solution to the problem of gaps in a new case's description is an approach that we call *constructive similarity assessment.* Constructive similarity assessment treats similarity assessment not as a simple comparison between fixed new and old cases, but as a process for deciding which types of features should be investigated in the new situation and, if the features are borne out by other knowledge, added to the description of the current case. Constructive similarity assessment does not merely compare new cases to old: using prior cases as its guide, it dynamically carves augmented descriptions of new cases out of memory.

Introduction

Case-based reasoning (CBR) systems facilitate processing of new cases by retrieving stored information about similar prior episodes, and adapting solutions from the prior episodes to fit the new situation (for a selection of current CBR approaches, see (Bareiss, 1991)). A fundamental issue in applying the CBR process is similarity assessment: how to judge the similarity between new cases and those retrieved from memory. The decisions of whether a retrieved case applies, and of where to adapt it if it fails to apply, depend on similarity judgements; consequently, similarity criteria have been the subject of considerable study. Many approaches have resulted (see (Bareiss & King, 1989) for a sampling), but they share a common property: they compare some subset of the features provided by the input case to features of cases stored in memory.

When input case descriptions contain all the information that is relevant to assessing the applicability of the new case, comparing features in the input case description to the features of old cases works well. However, for the task of case-based explanation construction during story understanding, the input cases presented to the understanding system will seldom provide sufficient information for feature comparisons to determine the relevance of prior cases. Consequently, case-based explanation requires not just comparing a static new case description to stored cases, but elaborating and expanding the new case's incomplete description.

Elaborating the new case requires seeking additional information about the current situation, either by inference from existing system knowledge or by investigation in the world. For example, a detective who knows nothing about person X and is informed of X*'s* death cannot hope to find an appropriate explanation by trying to remember the most similar previous episodes of death—the new case does not yet include sufficient information. Likewise, a story understander facing an anomalous situation is unlikely to begin with explicit knowledge of the important factors to consider during similarity assessment: the central problem for explanation is not matching fixed sets of features, but building up what the new case really is. Thus for both detective and story understander, the information provided by explicit inputs is likely to be too sparse for feature matching to be reliable.

It might appear that the problem of incomplete input case descriptions to a CBR component could be solved by preprocessing the new case. If the input case can be elaborated to select important features and fill in missing information, the case-based reasoning process can apply traditional similarity assessment procedures to the resulting case description. However, a circularity vitiates this method: the preprocessing phase of identifying the needed features would need to analyze the current situation in order to select the features to fill in, and that analysis of the current situation is the entire problem that the system originally needed to solve.

We propose an alternative approach: using stored cases themselves to guide case elaboration during the similarity assessment process. The goal of a case-based explanation system is to form a coherent view of the new situation; previously-explained cases can suggest features relevant to the new explanation, even if those features were omitted from the original case description. In this view, what is important is not the match between old and new features *per se*, but the ability of the old case to guide formation of a coherent view of the new situation. Here similarity assessment goes beyond comparison to become the constructive process of hypothesizing, from prior cases in memory, important features beyond the case description of the current situation, and attempting to build up a new candidate case description including those features. We call this approach *constructive similarity assessment*.

Constructive similarity assessment differs from traditional methods in two fundamental ways. First, rather than treating the features of an input case as fixed by the input case description, it treats the input description as a starting point for further elaboration. Second, the types of features of the new case that are elaborated depend on *suggestions* from the old case, but considerable adaptation may be required to fit new circumstances. Thus unlike straightforward feature matching, which compares features of a static new case description with prior cases, constructive similarity assessment uses the current contents of case memory as starting point for *deriving* features to consider as part of the new situation.

By using information in case memory to guide elaboration of an input case, constructive similarity assessment allows case-based reasoning systems to deal more flexibly and effectively with incomplete input cases and to better guide their search for additional information. The following sections expand on the process, showing how it extends the capabilities of a case-based reasoner to deal with poorly-defined or incomplete input cases, and sketching how the process has been investigated in the context of case-based explanation of anomalous events.

The Case Description Problem

In certain domains, input cases routinely provide all the information that needs to be considered during similarity assessment. For example, the standard input to a planner or problem-solver is a set of goals and constraints. Descriptions of those goals and constraints provide the essential information that CBR systems such as CHEF (Hammond, 1989) and JULIA (Kolodner, 1987) need in order to judge the similarity of prior cases stored in their memories. Likewise, in legal domains, input cases are routinely described in legal briefs that include all relevant features of the situation under consideration; they provide all the information that needs to be considered by CBR systems such as HYPO (Ashley & Rissland, 1987) and GREBE (Branting & Porter, 1991) as those systems identify similar cases. In such domains, for which input cases are guaranteed to include sufficient relevant features, traditional similarity assessment—comparison of the new case's features with features of a stored case—is appropriate.

However, input information is not always complete. For example, a lawyer taking on a case will probably not be content with the information initially provided, and will need to seek additional information. Likewise, it is crucial for story understanders that explain anomalous situations to build up the relevant details of the anomalous new case.

Stories are incomplete, and in principle, any of the many possible inferences from the text of a story could be relevant to explaining an anomaly, but forming all those connections is an overwhelming task (Rieger, 1975). For example, suppose an understanding system attempts to explain the breakdown of a car during a routine shopping trip. The features of the situation relevant to the breakdown's explanation may not be included in the story at all—even if the breakdown is caused by a ruptured hose, the story is unlikely to identify the hose's weakness before the fact.

Thus when explaining anomalies, adequate descriptions of input cases are hard to generate because no case description can include all the features of a real-world situation, and it may be impossible to identify *a priori* which factors of the stated situation to include in the description of the new case. In the example of the car breakdown, *a priori* schemes might suggest that the case should include features known to be directly related to the engine, such as engine noises, while omitting other features, such as which groceries were purchased. However, as the following mechanic's anecdote shows, unexpected aspects of cases may be important to their analysis:

> An elderly customer told a mechanic that her car's starting depended on the type of ice cream she bought: whenever she bought peppermint the car refused to start, but whenever she bought vanilla it started perfectly. The mechanic knew that peppermint could not affect the car's starting, but humored her with a test drive to buy ice cream. She parked at the store, went inside and bought a pint of vanilla; the car started perfectly. She drove around the block a few times, parked and bought peppermint. When she came out, the engine would not start. (Porter, 1973, pp. 253-254)

In this example, the flavor *was* implicated in the explanation: vanilla was sufficiently popular to be prepackaged, while peppermint was hand packed. Hand packing caused the purchase to take a few minutes more, allowing fuel from the carburetor to percolate into the engine and flood it. Because of the wide range of possible features, identifying important features of a new case is a difficult problem for case-based explanation.

Using Experience to Decide Case Features

Although it is impossible for a case-based explainer to determine important case features *a priori*, the case-based explanation framework suggests a method for deciding which features to consider adding to an initial case description. Because a case-based explainer attempts to use suggestions from prior explanations to explain the current case, it is natural to look for the additional features that are suggested by comparisons between the new case description and previous explanatory cases. If relevant features are not already present in the input description of the new case, the system can determine whether they apply by pursuing their connections to prior knowledge; if they apply, or if they suggest adapted features that apply, it can add the derived features to its description of the new case. Thus constructive similarity assessment uses retrieved cases to suggest which paths to pursue when elaborationing a current case, and the result is not just an evaluation that a prior case is relevant, but a new picture of the input case. This process is summarized in figure 1.

Constructive similarity assessment models our intuitive picture of the process that detectives, doctors or mechanics apply to deal with sketchy initial information. Rather than simply comparing old and new cases, they use old cases to guide their development of a picture of the new situation. For example, a detective hearing of the death of a millionaire might immediately be reminded of the previous murder of a millionaire by his heirs. If the only available information is that a millionaire died, the match between old and new cases is quite partial. However, the initial comparison shows that it is possible the old case is relevant, and suggests possible features to try to establish. If the previous millionaire was killed by hostile heirs, the detective looks for another feature in the current situation: did the dead millionaire have heirs who disliked him?

Features omitted from the input case may be built up in two ways: either by seeking external information, as a detective might do by interviewing the millionaire's acquaintances, or by inference based on the information already in the system's memory. In either case, the guidance of the prior case makes it possible to focus processing. If the decision of what to include in the description of a new case were made entirely before retrieval, the system would have no way to choose important features from the countless aspects of the situation that could theoretically be relevant. However, when the decision of what to include is made in light of a prior case, consideration is constrained to look at a much smaller set of features, and those features are guaranteed to give useful information: their presence or absence helps confirm or disconfirm the applicability of the prior case that the system is attempting to apply.

Literal Matching vs. Matching Adapted Features

Even after a retrieved case has suggested the features to consider in an input, constructive similarity assessment does not reduce to simple feature matching with the elaborated input case. Because the appropriateness of the retrieved case is determined entirely by its ability to suggest a coherent view of the new situation, much more abstract similarity relationships may apply. For example, if the anecdote connecting peppermint ice cream to starting problems were retrieved from memory to be applied to a new case, relevance of that story to a new situation would depend on whether the

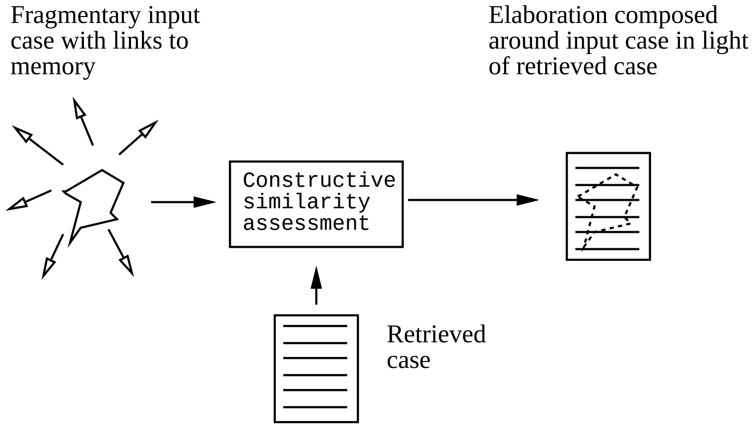

Figure 1: How constructive similarity assessment elaborates new cases based on experience.

new situation involved something *causally equivalent* to the ice cream purchases: delays correlated with an engine problem.

Thus unlike traditional similarity assessment methods, constructive similarity assessment does not require that the same predicates apply in order to consider two situations similar; what must match is the more abstract causal structure. Accordingly, similarity assessment need not actually evaluate the match between individual features of the old or new cases. What is important is whether, with guidance from the retrieved case, the features of the new case can be adapted into a coherent picture of the current situation. Because this method makes similarity assessment depend strongly on the ability to hypothesize elaborations of new cases and to judge the reasonableness of possible elaborations given current knowledge, its results depend on the contents of memory. In addition, the results depend on adaptation strategies for fitting an old case to a new situation: constructive similarity assessment treats a retrieved case as appropriate to the new situation, regardless of how dissimilar its features might be, if an adaptation of the retrieved case provides a suitable elaboration of the new situation.

Programs for Constructive Similarity Assessment

As a more concrete illustration, we consider the constructive similarity assessment process investigated in the case-based explanation framework of SWALE (Kass, 1986; Leake & Owens, 1986; Schank & Leake, 1989) and of ACCEPTER, a system which began as the case evaluation component of SWALE. (Leake, 1992). Both SWALE and ACCEPTER are story understanding systems that use case-based reasoning to explain anomalous events in news stories. The primary example of SWALE is the story of Swale, who was a superstar 3-year-old racehorse in peak shape, decisively winning major victories, who collapsed and died without warning a few days after winning the Belmont stakes. Anomalies processed by ACCEPTER also include the death of basketball star Len Bias the day after being first choice in the basketball draft, the explosion of the space shuttle Challenger, and the news that the American warship Vincennes shot down a civilian airliner.

For each of SWALE's and ACCEPTER's stories, inputs to the system are highly incomplete: They correspond to the information contained in newspaper headlines. In general, there will be many explanations in memory for an event such as a sudden death, and any of those explanations would match the few supplied features of the input case equally well. Both as a component of SWALE, and as a stand-alone system, the job of ACCEPTER is to guide constructive similarity assessment: it uses knowledge of likely events to evaluate elaborations of the input case in light of experience, to guide elaboration of input case descriptions based on prior cases.

Unlike most CBR approaches, the case-based explanation model does not treat matches between given features in old and new cases as necessarily important to using the new case: what is important is simply whether consideration of the old case gives rise to a (possibly quite different) coherent scenario for the new situation. For example, one of the explanations that SWALE retrieves for Swale's death is the explanation for the death of the rock star Janis Joplin: death from an overdose of recreational drugs taken to escape stress. Few features of this explanation match at a literal level, but the suggestion of drugs leads to a plausible alternative more directly connected to horses: that Swale might have died from an overdose of performance-enhancing drugs. Another retrieved explanation is the death of the runner Jim Fixx because jogging overtaxed a hereditary heart defect, which again fails to match at a literal level (horses are not joggers, and the input case provides no information about a heart defect). Nevertheless, that explanation leads to a reasonable hypothesis: Swale's racing overtaxed a heart defect to lead to his death.

The features implicated in these generated explanations (death from performance-enhancing drugs, or death from racing with a heart defect) are not part of the input case presented to SWALE, which include only Swale's age, the information about SWALE's Belmont victory and the fact that he died. Consequently, the resultant explanations might not receive high rankings by traditional similarity assessment. However, each explanation can be connected to known information about Swale to form a larger new case with which they share important features. For example, the use of performance-enhancing drugs fits stereotypes for horse racing, supporting that Swale's trainer might have given him a drug overdose; a heart defect follows from inbreeding, which might be associated with purebred animals. The two candidate explanations suggest that these links be pursued to form a more complete picture of the input case, either confirming or refuting those hypotheses. By following such links and considering their ramifications, constructive similarity assessment incrementally builds up a richer picture of a sketchy input case.

Relationship to Other Perspectives

Constructive similarity assessment is particularly relevant to three questions in CBR: what to include in a case representation, how changing circumstances affect criteria for similarity assessment, and how case features should be compared.

What to include in a case representation: Rather than requiring a new case to be completely specified in advance, constructive similarity assessment considers how a chosen candidate case can suggest inference of features that may not be included in the input case description. Owens (88) also considers how a case library can guide interpretation of unanalyzed situations, but from a different perspective: That work concentrates on how the contents of memory can guide selection of features to derive from an input case in order to discriminate between possibly-relevant stored cases.

How changing circumstances affect similarity assessment: Previous CBR research has proposed models of similarity assessment that are dynamic with respect to changing system goals for applying retrieved cases; in those models current goals determine which features to consider important in a given situation (Ashley & Rissland, 1987; Kolodner, 1989; Leake, 1991). The constructive similarity assessment process is dynamic in a different way: It elaborates new situations according to the current contents of memory—specific cases, general background knowledge and specific beliefs—and dynamically alters how it will understand a given new case, based on the current contents of memory.[1]

Comparison of case features: The constructive similarity assessment process we propose allows old cases to be applied to new situations not only if they match the facts of new situations, as in most similarity assessment criteria, but if their features are related on a much more abstract level. Both GREBE (Branting & Porter, 1991) and PROTOS (Bareiss, 1989) also go beyond requiring literal feature matches, using explanation of more abstract relevance of features to decide similarity (e.g., PROTOS can match the legs of one chair to the pedestal of another, because both serve as a seat support). Constructive similarity assessment, however, uses this process to suggest directions to investigate in order to suggest properties of the current case that may not be present in the initial representation of the new case. For ACCEPTER, old cases are applicable if adaptations of their features yield a picture of the new situation that fits other knowledge in memory. For example, the explanation of a recreational drug overdose is considered relevant because it can be adapted to suggest a *new* picture of the death that makes sense in light of stereotypes for horse racing: that Swale was drugged by his trainer.

Conclusion

Similarity assessment processes traditionally assume that all relevant information about a new

[1] In this respect, the flavor of this model is very similar to that of Schank's dynamic memory theory (Schank, 1982).

case is available at the time of case retrieval, so that similarity assessment is simply matching of given features. We have shown that this assumption does not hold in real-world explanation; nor will *a priori* methods work for elaborating the important features to consider in a new case. Consequently, a more flexible means of building new cases is needed. The solution we present is constructive similarity assessment. Rather than treating cases as having a fixed set of predetermined features, it treats them as starting points for further elaboration based on suggestions of the retrieved case; rather than evaluating only the match between the new situation and a fixed set of features in the retrieved case, it takes features in the retrieved case only as a starting point, allowing them to be adapted as long as the result is a plausible scenario given other system knowledge. This method guides processing of poorly-understood situations by combining sketchy input information with suggestions from memory to go beyond simple comparison and produce a sharper view of the events being understood.

Acknowledgment

I would like to thank Chris Riesbeck for helpful comments and suggestions leading to this paper.

References

Ashley, K. & Rissland, E. (1987). Compare and contrast, a test of expertise. In *Proceedings of the Sixth Annual National Conference on Artificial Intelligence*, pp. 273–284 Palo Alto. AAAI, Morgan Kaufmann, Inc.

Bareiss, R. (1989). *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning.* Academic Press, Inc., San Diego.

Bareiss, R. (Ed.)., Bareiss (1991). *Proceedings of the Case-Based Reasoning Workshop*, Palo Alto. DARPA, Morgan Kaufmann, Inc.

Bareiss, R. & King, J. (1989). Similarity assessment in case-based reasoning. In Hammond, K. (Ed.), *Proceedings of the Case-Based Reasoning Workshop*, pp. 67–71 San Mateo. DARPA, Morgan Kaufmann, Inc.

Branting, K. & Porter, B. (1991). Rules and precedents as complementary warrants. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pp. 3–9 Anaheim, CA. AAAI.

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

Kass, A. (1986). Modifying explanations to understand stories. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* Amherst, MA. Cognitive Science Society.

Kolodner, J. (1987). Extending problem solving capabilities through case-based inference. In *Proceedings of the Fourth International Workshop on Machine Learning* Irvine, CA. Machine Learning, Morgan Kaufmann.

Kolodner, J. (1989). Selecting the best case for a case-based reasoner. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pp. 155–162 Ann Arbor, MI. Cognitive Science Society.

Leake, D. (1991). ACCEPTER: a program for dynamic similarity assessment in case-based explanation. In Bareiss, R. (Ed.), *Proceedings of the Case-Based Reasoning Workshop*, pp. 51–62 San Mateo. DARPA, Morgan Kaufmann, Inc.

Leake, D. (1992). *Evaluating Explanations: A Content Theory.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Leake, D. & Owens, C. (1986). Organizing memory for explanation. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 710–715 Amherst, MA. Cognitive Science Society.

Owens, C. (1988). Domain-independent prototype cases for planning. In Kolodner, J. (Ed.), *Proceedings of a Workshop on Case-Based Reasoning*, pp. 302–311 Palo Alto. DARPA, Morgan Kaufmann, Inc.

Porter, J. (Ed.). (1973). *The Family Car.* Time-Life Books, New York.

Rieger, C. (1975). Conceptual memory and inference. In *Conceptual Information Processing.* North-Holland, Amsterdam.

Schank, R. (1982). *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press, Cambridge, England.

Schank, R. & Leake, D. (1989). Creativity and learning in a case-based explainer. *Artificial Intelligence*, *40*(1-3), 353–385. Also in Carbonell, J., editor, *Machine Learning: Paradigms and Methods*, MIT Press, Cambridge, MA, 1990.