Relevance in plan recognition for advice-giving systems

Robin Cohen
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
rcohen@dragon.uwaterloo.ca

Ken Schmidt and **Peter van Beek**
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
vanbeek@cs.ualberta.ca

Overview

We have done work in the area of plan recognition. Traditionally, plan recognition systems operate by matching observations (typically, actions of some user) onto a pre-defined plan library, which serves to record all plans in the domain of application. Advice-giving systems may benefit from plan recognition, determining the user's plan and possible faults in the plan in order to produce an improved response to the user. For this task, most plan recognition systems have heuristically determined a best guess for the plan of the user. The problem of relevance arises, as plan libraries may become large and observations may be impoverished requiring a selection of the plan library for focus.

We have first identified those cases where plan ambiguity is not necessary to resolve, in order to present a response (same response results, regardless of specific user plan). We then propose to involve the user, engaging in clarification dialogue, when ambiguity matters. We try never to overcommit, but to have users continue to provide more information, to help detect the most relevant paths. The challenge is to develop algorithms for the generation of clarification dialogues which serve to uncover the user's plan (to the point where ambiguity does not matter to the formulation of a response) through an exchange which is both brief and clear. We have investigated strategies for reducing the length of clarification dialogues, by carefully selecting nodes from the plan library to ask the user about. As the user provides responses to questions being generated, the plan library is pruned to reflect only possible plans of the user.

(Ir)relevance arises in our framework in two separate phases; hence, the treatments differ slightly. In the context of plan recognition, we treat as irrelevant a more specific formulation of the user's plan, when the current (vaguer) representation is sufficient for generating an appropriate response. In the context of selecting a node of the plan library to ask the user about in clarification dialogue, we treat as most relevant those nodes which will lead to a briefer clarification dialogue with the user, i.e. those which serve to prune the plan library to the point where the plan of the user is determined, at the level of detail required for generating the appropriate response to the user. The definitions are therefore: irrelevant nodes in a knowledge base are those which do not clarify further the response being generated for a user, while the most relevant nodes are those which generate the appropriate response with the least computational effort (and, for natural language, with coherence and clarity retained, for user comprehension).

Further details

In this section, we sketch very briefly our proposed solutions for (i) determining that the output of the plan recognition system does not require further refinement (i.e. that further specification is irrelevant to the task at hand) and for (ii) attempting to determine the user's plan, through plan recognition, by considering the most relevant parts of the plan library.

Conditions for stopping

As discussed at greater length in (van Beek and Cohen 91, van Beek et al. 93), it is not necessary to resolve possible ambiguity about a user's plan if the same response would be generated for a user (in an advice-giving setting), regardless of further specification of the user's plan. Since advice-giving systems should be as cooperative as possible (Joshi et al. 84), it is important to determine the user's plan and potential faults with that plan, and then to discuss the potential faults in the response which is generated.

An example will serve to illustrate. Consider a course-advising system for students. User1 is a Science student. User2 is an Arts student. Suppose each one asks: "Can I enroll in CS102 next term?". CS102 is a course restricted to Arts students. The system's response to to User1 is "No" and further specification of his plan is not necessary. This would be termed a fault of the class "Failure of preconditions" and would apply, regardless of whether User1 was interested in enrolling to satisfy program requirements, to avoid a heavy course, to keep all his courses in the morning, etc. - i.e. regardless of further specification of his underlying plan. The response to User2 may require further clarification.

We therefore propose a procedure which critiques all the possible plans of the user, labels them all according to potential faults and determines whether the possible

plans all belong to the same fault class. If so, further disambiguation is not necessary. If there is ambiguity, we propose to resolve it via a clarification dialogue with the user, discussed below.

Selecting relevant nodes

We have employed a hierarchically organized plan library and the algorithms of (Kautz 87) in our implementations. Our basic strategy for selecting nodes from the plan library to query the user about, during clarification, is one of top-down traversal. A disjunctive event is selected to ask the user about and, based on the user's response, the set of possible plans is reduced to include only those plans which contain the event (a "yes" response) or those which do not (a "no" response"). See (Schmidt 94, Cohen et al. 94) for further details.

This clarification procedure can be improved, by incorporating several strategies which curtail or avoid the basic top-down traversal. These strategies are designed to direct the selection of nodes for the clarification queries, to those nodes which are more likely to be relevant.

Our "catalogue" of strategies includes the following: (i) when there are several nodes at the same level and one must be selected to ask the user about, allow for likelihoods (ib) in the same situation as (i), if no likelihoods are available, select the node based on a determination of the number of fault partitions which may be eliminated by a "no" reply (where a fault partition includes all plan with the same fault label) (ii) when there are several plans of the same fault partition, key nodes may be selected, if a pre-processing stage indicates that these nodes will critically eliminate all plans in a fault partition. (iii) when there is a serious fault in the user's plan, to the extent that further clarification (though necessary, in the sense that it changes the possible response to to the user) is not valuable (rather, repairing the fault is critical), we allow for early termination of the plan library top-down traversal (iv) when the plan library contains "diamond-shaped" sections, it may be the case that all intervening nodes, once clarified, simply lead back to the same node which requires further clarification. In this case, the intervening dialogue does not take us closer to our goal of critically disambiguating the user's plan. We develop criteria for determining when to skip branch points.

We offer one brief illustration, for the strategy of employing key events. Key nodes are identified as those which are shared by all plans of a particular partition and not shared by plans in other partitions. (Using a simulated example for courses at the University of Alberta), suppose there are 40 possible plans for programs of study and user asks "Should I enroll in CMPUT 165?". Suppose further that based on this user's profile, the 40 plans are partitioned into two fault partitions. Partition-1 contains two faultless plans (Honours and Specialization Computing Science) and Partition-2 contains the 38 other plans, each of which has the same fault annotation with respect to CMPUT-165. A top-down traversal of the plan library may ask the user which kind of Bachelor's degree he is planning to pursue, then which area he is planning to concentrate in (e.g. physical sciences, mathematical sciences, etc.), then which department, finally leading to the disambiguating node of computer science. A strategy which employs key events would ask directly about computer science first, and disambiguate the user's plan to the point where an appropriate response can be generated, using only one question.

In short, the introduction of key events, tied to the tracking of fault partitions, allows irrelevant questions to be bypassed, resulting in a briefer (and hopefully more friendly) clarification dialogue.

Closing remarks

A more general position underlying our approach for focusing on relevant information in the knowledge bases of automated reasoning systems as follows: try to get users more involved in the process. This is a point of view which we first expressed in (Cohen et al. 91). For our application area of plan recognition, we have illustrated how users may be engaged in clarification dialogues, to deepen the specification of the input to the automated reasoning system. But it may also be feasible to bring users back into the loop to improve the representation of the underlying knowledge base. Allowing automated reasoning systems to pause and ask for further assistance would be a worthwhile design strategy.

References

Cohen, R., Spencer, B. and van Beek, P.; In search of practical specifications - allowing the user a more active role; AAAI Spring symposium on implemented KR and reasoning systems, March 1991.

Cohen, R., Schmidt, K. and van Beek, P.; A framework for soliciting clarification from users during plan recognition; Proceedings of UM-94 (user modeling conference), 1994.

Joshi, A., Webber, B. and Weischedel, R.; Living up to expectations: computing expert responses; AAAI-84, p.169-175.

Kautz, H.; A Formal Theory of Plan Recognition; PhD thesis, University of Rochester, 1987. Available as: Dept. of Computer Science Tech Report 215.

Schmidt. K.; Clarification dialogues for plan recognition in advice-giving settings; M.Sc. thesis, Dept. of Computing Science, University of Alberta, 1994.

van Beek, P. and Cohen, R.; Resolving plan ambiguity for cooperative response generation; IJCAI-91.

van Beek, P., Cohen, R. and Schmidt, K.; From plan critiquing to clarification dialogue for cooperative response generation; *Computational Intelligence*, **9**, p.132-154, 1993.