WPI Computer Science Department
Spring Semester 2000

CS 538 CHEF Expert Systems

Prof. Dave Brown
Office, Fuller Lab. 131
Phone: (508) 831-5618
Email: DCB@CS.WPI.EDU

EXPERT SYSTEM PROFILE

GENERAL

Domain:

Szechwan cooking

Main General Function:

Case-Based Planning

System Name:

CHEF

Dates:

1987

Researchers:

K. Hammond

Location:

University of Chicago, Dept. of Computer Science/Psychology

Language:

UNKNOWN

Machine:

UNKNOWN

Brief Summary:

CHEF is a case-based planner, which can output new recipes given particular ingredients and tastes. Recipes are viewed as plans. They provide the sequence of steps that must be carried out to achieve the creation of some dish. CHEF creates its plans by recalling old plans that worked under similar circumstances and modifying them to fit its new situation. CHEF learns 1) new plans 2) the features that predict failures 3) past repairs to faulty plans. This learning is accomplished by saving the different results of the planner's own experiences.

Related Systems:

The first system that might be called a case-based reasoner was the CYRUS system, developed by Janet Kolodner [Kolodner-83], at Yale University (Schank's group). CYRUS was based on Schank's dynamic memory model and MOP theory of problem solving and learning [Schank-82]. It was basically a question-answering system with knowledge of the various travels and meetings of former US Secretary of State Cyrus Vance. The case memory model developed for this system has later served as basis for several other case-based reasoning systems (including MEDIATOR [Simpson-85], PERSUADER [Sycara-88], CHEF [Hammond-89], JULIA [Hinrichs-92], CASEY [Koton-89]).

CATEGORY TWO

Characterization of Givens:

Old plans are stored in abstraction hierarchy semantic net. Causal rules, and repair strategies were built into the system. Chef’s input is a set of goals for different tastes, textures, ingredients, and type of dish.

Characterization of Output:

The planner learns 1) new plans; 2) the features that predict failures; 3) past repairs to faulty plans that it can reuse. CHEF outputs new recipes with particular ingredients and tastes. Recipes are viewed as plans. They provide the sequence of steps that must be carried out to achieve the creation of some dish.

Characterization of Data:

The data is reliable but not complete. Previous plans are organized as abstraction hierarchy, and can be indexed by features and the problems that they avoid. All the (Thematic Organization Packets**)** TOPs are associated with causal configurations that lead to failures and strategies for fixing them.

Generic Tasks:

Its main task is to build new recipes on the basis of users requests. One subtask is the plan retrieving using classification. Another important subtask is the plan repairing, which uses backtrack reasoning, or causal explanation.

Theoretical Commitment:

CHEF proposes a new approach to planning that allows it to avoid many of the problems of more traditional planning systems. Some of its advances are based on the fact that it reuses plans. Some are the result of the fact that it incrementally repairs its plans and learns from experience.

Case repair involves detecting the errors of the current solution and retrieving or generating explanations for them. The best example is the CHEF system, where causal knowledge is used to generate an explanation of why certain goals of the solution plan were not achieved. [Case based reasoning: A. Aamodt, E. Plaza (1994)]

Reality:

The evaluation task takes the result from applying the solution in the real environment. This is usually a step outside the CBR system. CHEF use a simulation program, where a solution (i.e. a cooking recipe) is applied to an internal model assumed to be strong enough to give the necessary feedback for solution repair.

CATEGORY THREE

Completeness:

UNKNOWN

Use:

UNKNOWN

Performance:

UNKNOWN

CATEGORY FOUR

Phases:

The system is organized into 6 distinct phases:

1. Problems Anticipation
2. Plan Retrieving
3. Plan Modification
4. Simulating Execution
5. Plan Repairing
6. Assign failure predictors

Subfunctions:

One subtask is the plan retrieving using classification. Another important subtask is the plan repairing, which uses backtrack reasoning, or causal explanation.

Use of Simulation or Analysis:

CHEF uses a simulation program, where a solution (i.e. a cooking recipe) is applied to an internal model to give the necessary feedback for solution repair.

System/Control Implementation Architecture:

Semantic net for storing old plans.
Production rules for causal analysis.

CATEGORY FIVE

Characterization of Structure Knowledge:

CHEF mainly uses hierarchy knowledge and causal knowledge.

Characterization of Process Knowledge:

Classification, Simulation and Backtrack reasoning

Deep or Surface:

CHEF uses Deep and Surface knowledge at the same time. When in the plan retrieval phase, it uses surface knowledge, and in the simulation and repairing phases, it uses deep knowledge.

CATEGORY SIX

Search Space:

The search spaces for CHEF are the old plans and repair strategies. The states are partially matched plans. CHEF use modification phase and repairing phase to do solution refinements. The space is decided by previous experiences.

Space Traversal:

Gradual refinement.

Search Control Strategy:

The search strategy is unique to CHEF system

Standard Search Strategies:

When a case solution generated by the reuse, (1) evaluate the case solution generated by reuse. If successful, learning from the success, (2) otherwise repair the case solution using domain-specific knowledge.

Search Control Characterization:

Knowledge-based, best-match

Subproblems:

CHEF checks if there is a partial match between the target goal and an object in the existing recipe. If the target goal and the existing object are under a common abstraction hierarchy, they are partial match. The subproblems are not independent. Refinement is needed.

Search Control Representation:

CHEF stores plans indexed by the goals they satisfy and the problems they avoid, and retrieve them by these indexes.

Search Control Strength:

It is very domain dependent and knowledge-full (a "strong" method).

CATEGORY SEVEN

Failure Method:

Case repair involves detecting the errors of the current solution and retrieving or generating explanations for them. In CHEF system causal knowledge is used to generate an explanation of why certain goals of the solution plan were not achieved. CHEF learns the general situations that will cause the failures using an explanation-based learning technique. This is included into a failure memory that is used in the reuse phase to predict possible shortcomings of plans. This form of learning moves detection of errors in a post hoc fashion to the elaboration plan phase were errors can be predicted, handled and avoided. A second task of the revision phase is the solution repair task. This task uses the failure explanations to modify the solution in such a way that failures do not occur. For instance, the failed plan in the CHEF system is modified by a repair module that adds steps to the plan that will assure that the causes of the errors will not occur. The repair module possesses general causal knowledge and domain knowledge about how to disable or compensate causes of errors in the domain. The revised plan can then be retained directly (if the revision phase assures its correctness) or it can be evaluated and repaired again.

Uncertainty:

UNKNOWN

Management of Uncertainty:

No Uncertainty Management.

Management of Time:

Time should be included in the causal knowledge.

CATEGORY EIGHT

Knowledge Representation Method:

Cases, Rules and Semantic nets.

Knowledge Representation Generality:

UNKNOWN

Knowledge Structuring:

CHEF uses causal knowledge to do simulation and backtracking, uses semantic nets to do partial match. The semantic nets correspond to the food hierarchy.

CATEGORY NINE

Alternative Representations:

No

Alternative Solution Methods:

No

Optimization:

The system always produces the best answer.

Multiple Results:

No

CATEGORY TEN

Interaction:

Chef’s input is a set of goals for different tastes, textures, ingredients, and type of dish.

Data collection:

CHEF requires all the requirements from users before execution; otherwise the recipe cannot be made up.

Data format:

The data is given in Natural Language.

Acquisition:

The system must have a way of acquiring knowledge from the expert user. Besides old plans, CHEF needs to know food abstract hierarchy for plan retrieval, causal rules for simulation and backtracking, TOPs for repairing, which CHEF cannot learn incrementally.

Learning:

CHEF learns the following

1. New plans;
2. Features that predict failures;
3. Past repairs to faulty plans

from it's own experience:

Explanation:

1. CHEF stores new plans indexed by the goals they satisfy and the problems they avoid. In storing a plan, it does not generalize the plan itself. It instead generalizes the features that are used to index the plan in memory.
2. When a failure occurs, the planner records a link between features of goals and the failure that they have caused. These memories of failures can be used when the goals arise again. When multiple features are required to predict a failure, all of them are linked to the memory of the failure. This memory is not activated unless all of the linked features are present.
3. Aside from storing plans and failures, a case-based planner also stores some of the repairs in the form of critics, indexed by the problems that they solve. These repairs are used when a planner predicts a problem, but cannot find a plan of the proper type to deal with that problem.

CATEGORY ELEVEN

Strengths:

Case Revision
When a case solution generated by the reuse phase is not correct, an opportunity for learning from failure arises. This phase is called case revision and consists of two tasks: (1) evaluate the case solution

generated by reuse. If successful, learning from the success, (2) otherwise repair the case solution using domain-specific knowledge.

Weaknesses:

UNKNOWN

----

profile.txt