

A Knowledge-Intensive Method for Conversational CBR

Mingyang Gu and Agnar Aamodt

Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7-9, N-7491, Trondheim, Norway
{Mingyang.Gu, Agnar.Aamodt}@idi.ntnu.no

Abstract. In conversational case-based reasoning (CCBR), a main problem is how to select the most discriminative questions and display them to users in a natural way to alleviate users' cognitive load. This is referred to as the question selection task. Current question selection methods are knowledge-poor, that is, only statistical metrics are taken into account. In this paper, we identify four computational tasks of a conversation process: *feature inferencing*, *question ranking*, *consistent question clustering* and *coherent question sequencing*. We show how general domain knowledge is able to improve these processes. A knowledge representation system suitable for capturing both cases and general knowledge has been extended with meta-level relations for controlling a CCBR process. An "explanation-boosted" reasoning approach, designed to accomplish the knowledge-intensive question selection tasks, is presented. An application of our implemented system is illustrated in the car fault detection domain.

1 Introduction

The basic idea underlying case-based reasoning (CBR) is to reuse the old solution to the previous most similar problem in helping solve the current problem. Before we can reuse any existing solution, we have to find the most similar previous problem, corresponding to the retrieve phase in the standard CBR cycle [5].

In the traditional CBR process, users are assumed to be able to give a well-defined problem description (a new case), and based on such a well-defined description a CBR system can find the most appropriate previous case. But this assumption is not always realistic. Users usually only have vague ideas about their problems when beginning to retrieve cases, and often describe them by surface features, while the previous cases have been described by providers using the essential features. Furthermore, even if users understand what their problems are and what aspects they should describe, they do not know exactly what terms to use to express their problems.

In general, the knowledge gap between case users and case providers is a major cause for the difficulty of case retrieval. Users usually input a problem description by "guessing" the appropriate feature terms, and the system either returns too many matched cases or none. Conversational Case-Based Reasoning (CCBR) [6] has been proposed to bridge this knowledge gap.

Conversational CBR provides a mixed-initiative dialog for guiding users to refine their problem descriptions incrementally through a question-answer sequence. In the CCBR process, a user's initial problem description is used to retrieve the first set of candidate cases. Subsequent questions, prompted by the CCBR system, will cut down this case set iteratively until a manageable number of cases remain. That is, instead of letting a user guess how to describe her problem, CCBR discovers a sequence of discriminative questions, which help to extract information from the user, and to construct the problem description automatically and incrementally. CCBR applications have been successfully fielded, e.g., in the troubleshooting domain [11, 16] and in the products and services selection in E-Commerce [23].

A core research concern in conversational CBR is how to minimize the cognitive load demanded on users to retrieve their desired cases [23, 22], which requires to select the most discriminative questions [6, 8, 9] and ask them in a natural way in the conversation process [8, 12].

Up to now, several methods, such as the static decision tree [10], the information gain metric [11, 13, 23], the occurrence frequency metric [6], the information quality metric [9], the similarity variance metric [21], and the attribute-selection strategies [20], have been proposed to support question selection in the conversational CBR process. However, all the methods mentioned above are basically knowledge-poor, that is, they only take statistical information into account. The potential that general domain knowledge has for playing a positive role in the question selection process is little explored. For example, if the answer to question B can be inferred from that of question A, or the answer to question A is easier or cheaper to obtain than that to question B, question A should be prompted to users before question B. Such a knowledge-intensive question selection approach can select and display discriminative questions based on their semantic relations rather than only their statistical metrics.

We have identified four tasks in conversational CBR, for which general domain knowledge has a potential to control and improve the process: feature inferencing, question ranking, question clustering, and question sequencing.

Feature Inferencing (FI). If one feature of a problem can be inferred from the current problem description, this feature can be added to the problem description automatically, instead of posing a question to the user. Users are likely not to trust a communicating partner who asks for information that is easy to infer. General domain knowledge (domain rules or domain models) can be used to infer the features implicit in the problem description.

Question Ranking (QR). In the conversation process, the identified discriminative questions need to be ranked intentionally before displaying them to users. An integrated method should be adopted, which uses not only the superficial statistical metrics of the questions, but also the semantic relations among them. For example, if the answer to question C can be inferred from one of the possible answers to question D, it may be better to ask question D first.

Even though an integrated question ranking module outputs a set of sorted questions, their screen arrangement and questioning sequence should not be decided by such a sorted order alone. The main reason lies in that people always hope to inspect or answer questions in a natural way. They would prefer to see a set of questions that are connected by some semantic relations, grouped together, and to

answer them in an uninterrupted sequence. These requirements are captured by the following two tasks:

Consistent Question Clustering (CQC). The arrangement of questions on the screen should be consistent, that is, the questions with some semantic relations among them should be grouped and displayed together, and the order of the questions in each group should be decided intentionally. For example, the questions having dependency relations among them should be grouped and displayed together.

Coherent Question Sequencing (CQS). The questions asked in the sequential question-answer cycles should be as related as possible, that is, the semantic contents of two sequential questions should avoid switching too often. For example, if in the previous question-answer cycle a more general question in an abstraction taxonomy is asked, the downward more specific question should be asked in the succeeding cycle rather than inserting other non-related questions between them.

The suggested knowledge-intensive conversational CBR process is illustrated in Fig. 1. The lines in bold are the modules used to complete the tasks identified above.

```

NewCase := New-Case-Formalize(InitialProblemDescription);
SequentQuestions := null; // (CQS)
Repeat:
  NewCase := Feature-Inference(NewCase); // (FI)
  SortedRetrievedCases := CBR-Retrieve(NewCase);
  DiscriminativeQuestions := Question-Identify(SortedRetrievedCases, NewCase);
  RankedDiscriminativeQuestions := Integrated-Question-Rank(DiscriminativeQuestions); // (QR)
  RankedDiscriminativeQuestions :=
    Ranked-Questions-Adjust (SequentQuestions, RankedDiscriminativeQuestions); // (CQS)
  GroupedRankedDiscriminativeQuestions := Question-Group(RankedDiscriminativeQuestions); // (CQC)
  Display(GroupedRankedDiscriminativeQuestions, SortedRetrievedCases);
  If (users find their desired cases or have no question to answer) then
    Return SelectedCases;
  Else
    SelectedQuestionAndAnswer := User-Select-and-Answer-Question();
    SequentQuestions := Sequent-Question-Identify(SelectedQuestionAndAnswer); // (CQS)
    NewCase := NewCase-Update(NewCase, SelectedQuestionAndAnswer);
  End If

```

Fig. 1. The knowledge-intensive CCBR process

In this paper we present an explanation-boosted reasoning approach for support of knowledge-intensive question selection. The use of explanation in case-based reasoning is not new, but the meaning of the term differs. In our approach, the explanation part of the process mainly uses general domain knowledge (rather than specific cases), targeted at system internal reasoning (rather than user understanding). However, the explanations constructed can also be displayed to the user for transparency, justification, and increased understanding. What we mean by explanation-boosted reasoning is a particular method for constructing explanation paths that exploit general domain knowledge for the question selection tasks. The method was briefly introduced in an earlier workshop paper [14], in which only two of the four question selection tasks were described. In the presented paper we extend the description to cover more CCBR tasks, we explicitly relate the tasks with meta-level relations for reasoning, and we present the first implemented version of the system.

The rest of this paper is organized as follows. In Section 2, we identify several semantic relations related to question selection. In Section 3, our explanation-boosted question selection method is described from the perspectives of knowledge representation, explanation construction and reasoning method. The system implementation of this approach, and related research, are summarized in Section 4 and Section 5, respectively. Our conclusion is drawn in Section 6.

2 Semantic Relations for Question Selection

General domain knowledge enables question selection to be based on semantic rather than purely syntactic criteria. Below, we describe a set of semantic relations among features, which influence question selection.

- Feature Abstraction. A feature can be described at different abstraction levels that form a subsumption hierarchy. The lower the level a feature belongs to, the more specifically it can describe the case, but the more difficult it will be to obtain. The appearance of a lower level feature can be used to infer the existence of higher level features. For instance, the feature of “Fuel Transmission Faulty” is a lower level feature than that of “Fuel System Faulty”. In [17], Gupta argued that the conversations should follow a downward taxonomic traversal to extract questions from general to specific, which prunes questions deemed irrelevant or implicitly inferred by the taxonomy. Here, we define a relation “subclass of” to express the relation of “feature abstraction”. “A is a subclass of B” means A is a lower level feature than B.
- Dependency Relations. A dependency relation between two features exists if the appearance of one feature depends on the existence of the other. For instance, the assertion that the fuel pump can pump fuel depends on that the car has fuel in its fuel tank. We define a relation “depends on” to describe dependency relations. “A depends on B” means B is a necessary condition for A.
- Causality Relations. The causality relation means that one feature can cause the occurrence of another feature. For example, an electricity system fault in a car can cause its engine not to start. Here, we define a relation “causes” to express causality relations. “A causes B” means B is the result of A.
- Co-occurrence Relations. A particular relation, “co-occurs with”, is defined to express that two features happen together, even though we cannot tell which one causes the other.
- Answer Acquisition Costs. The costs or difficulties of obtaining answers to different questions are various [11]. For instance, to test whether a switch has a fault is more difficult than to test whether the battery has electricity. The relation “is more costly than” is defined to represent that the answer to one question is more difficult or costly to obtain than the answer to another question.

How the above relations can be used to support the knowledge-intensive question selection tasks is illustrated in Table 1.

Our intention here is not to enumerate all the semantic relations that influence the question selection in conversational CBR, but to give some examples and illustrate how our approach can utilize them to improve the question selection process. System

implementors can also define their own semantic relations which they think influence the question selection process. We will show that it is straightforward to add a new semantic relation into the question selection application later in the paper.

Table 1. Semantic relations used in the knowledge-intensive question selection

	Feature Inferencing	Knowledge-Intensive Question Ranking	Consistent Question Clustering	Coherent Question Sequencing
Feature Abstraction (A is a subclass of B)	Inference B from A	Ask A after B	Group A and B together	A succeeds B
Dependency Relations (A depends on B)	Inference B from A	Ask A after B	Group A and B together	A succeeds B
Causality Relations (A causes B)	Inference B from A	Ask B after A	Group A and B together	
Co-occurrence Relations (A co-occurs with B)	Inference B from A; Inference A from B		Group A and B together	
Answer Acquisition Costs (A is more costly than B)		Ask A after B		

3 An Explanation-Boosted Question Selection Approach

In this section, the explanation-boosted question selection approach is described, focusing on three architectural and methodological issues: knowledge representation, explanation construction, and explanation-boosted reasoning method.

3.1 Knowledge Representation

A frame-based knowledge representation model, which is a part of the CREEK system [1, 3, 24], is adopted in our system. In CREEK, both case-specific knowledge and general domain knowledge are captured as a network of concepts and relations, each concept and relation is represented as a frame in a frame-based representation language. A frame consists of a set of slots, representing relationships with other concepts or with non-concept values, e.g. numbers. A relationship is described using an ordered triple $\langle C_f, T, C_v \rangle$, in which C_f is the concept described by this relationship, C_v is another concept acting as the value of this relationship (value concept), and T designates the relation type, simply called relation. The equation $T = C_v$ can also be used to describe a relationship when C_f is default. Viewed as a semantic network, a concept corresponds to a node and a relation corresponds to a link between two nodes.

In the system presented here, knowledge is represented at two levels. The first is the object-level, in which case-specific knowledge and general domain knowledge are represented within a single representation framework. The second is the meta-level, which is used to express the inter-relations of the semantic relations influencing the question selection tasks.

3.1.1 An Object-Level Knowledge Representation Model

As an illustration of how a case is described, Fig. 2 shows, in a frame view, the contents of a new case in the car fault domain, while Fig. 3 shows, in a semantic network view, a part of the integrated knowledge base for that domain. As can be seen, the semantic relations identified in Section 2 are represented as relations connecting different concepts. Cases are integrated into the general domain model, since all case features are defined as concepts within it.

The relationship values, which have corresponding relationships in the retrieved cases, but do not have the same type relationships in the new case, can be converted into discriminative questions. For example, if the relationship value, “Engine Dose Not Turn”, has a relationship in one of the retrieved cases, that is, “has engine status = Engine Does Not Turn”, but does not have the same type relationship in the new case, then a discriminative question, “What is the engine status of your car?”, is added to the discriminative question list.

We define a function that maps a set of relationship values to a set of questions, Q : relationship value set \rightarrow question set. On this function, we define the following properties:

- The question transformed from one relationship value is the same as those formed by the relationship values that belong to the same relation type. So we only predefine one question for each relation, which is shared by the relationship values belonging to this relation. For example, $Q(\text{“Engine Fires”}) = Q(\text{“Engine Turns”}) = Q(\text{“Engine Does Not Fire”}) = Q(\text{“Engine Stops After A Few Seconds”}) = \text{“What is the engine status of your car?”}$.
- The semantic relations that exist between two relationship values are transferred to the two questions transformed by these two relationship values. For instance, the “causes” relation that “Fuel Pump Damaged” “causes” “Engine Stops After A Few Seconds” is transformed to $Q(\text{“Fuel Pump Damaged”}) \text{ “causes” } Q(\text{“Engine Stops After A Few Seconds”})$. Following the “has question” link to the actual question, “What is the fuel pump status of your car?”, it follows that this question “causes” the question “What is the engine status of your car?”.

The screenshot shows a software interface for defining a case. It has several sections:

- Name:** A text box containing "Case 8".
- Description:** A text box containing "Case 8 is a new case to show the knowlege-intensive conversational retrieve process."
- Case status:** A dropdown menu with the selected option "_unsolved case".
- Case Type:** A dropdown menu with the selected option "Car Starting Case".
- Table:** A table with two columns: "Relation-type" and "Value".

Relation-type	Value		
has battery status	Battery Ok		
has engine status	Engine Does Not Fire		
has weather condition	Damp Weather		
- Footer:** Two dropdown menus labeled "<Choose relation>" and "<Choose value>" followed by an "Add" button.

Fig. 2. The frame structure for a car starting case in CREEK

3.1.2 Meta-Level Relations and Reflective Reasoning

Four meta-level relations have been defined in order to control the inference processes related to each of the four question selection tasks. For feature inferencing, we define the “*infers*” relation to express that if A infers B, we can get B from the existence of A. This relation has the property of transitivity that if A infers B and B

infers C then A infers C. Several semantic relations identified in Section 2, “subclass of”, “depends on”, “causes” and “co-occurs with” are subclass relations of the “infers” relation since all these relations can be used to infer the existence of the post-condition based on the appearance of the pre-condition.

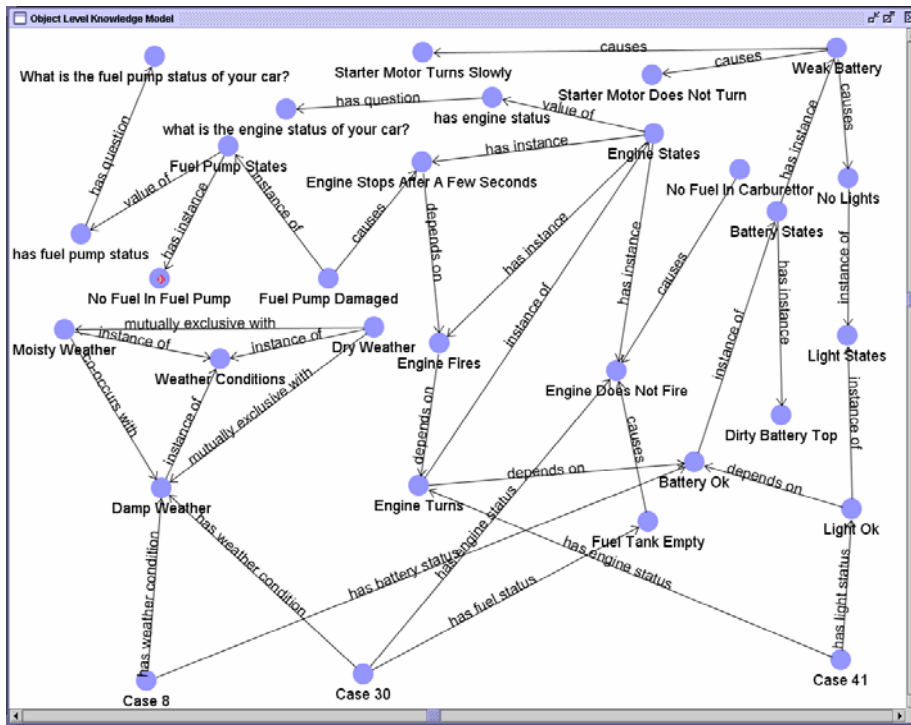


Fig. 8. The “Dialogue” pane in one conversation session

The second metal-level relation, “*appears after*”, is defined to complete the question ranking task. “A appears after B” means that Q(A) should be asked after Q(B). This relation also has the property of transitivity that if A appears after B and B appears after C then A appears after C. We define several relations identified in Section 2, “subclass of”, “depends on”, “caused by” and “is more costly than” as the subclass relations of the “appears after” relation because all these relations can rank the pre-condition question to be asked after the post-condition question.

The third meta-level relation, named “joins”, is defined to realize the consistent question clustering task. “A joins B” means that Q(A) should be grouped and displayed together with Q(B). We define several relations identified in Section 2, “subclass of”, “depends on”, “causes” and “co-occurs with” as subclass relations of the “joins” relation because all the questions connected by these relations should be grouped and displayed together. The transitivity property is not defined on “joins” because we assume that only the questions that have direct “joins” relations between them can be grouped and displayed together.

The last meta-level relation, called “succeeds”, is used in the coherent question sequencing task. “A succeeds B” means Q(A) should be asked directly after Q(B) in

two sequential question-answer cycles. There are two relations, “subclass of” and “depends on”, defined as the subclass relations of this “succeeds” relation. On this basic relation, the transitivity property is also defined, that is, if A succeeds B and B succeeds C, we can get A succeeds C.

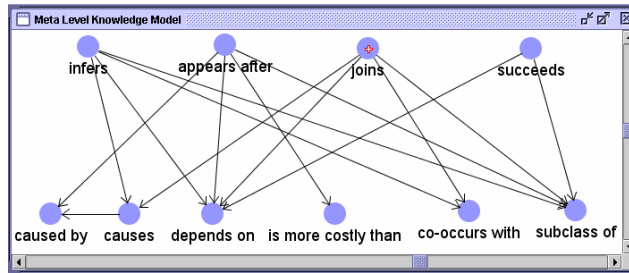


Fig. 4. The structure of the meta-level knowledge representation model

Fig. 4 shows the structure of the meta-level knowledge representation model described above. The top part relations are the meta-level relations defined above, while the bottom part relations are the semantic relations identified in Section 2. The lines from the top part relations to the bottom part relations designate the “has subclass” relations, while the line from “causes” to “caused by” is a “has inverse” relation.

One type of reflective reasoning operation, subclass inheritance, is made explicit in this meta-level knowledge representation model. Subclass inheritance is a special case of the more general “plausible inheritance” mechanism in CREEK [1], and makes subclass relations inherit the properties and reasoning operations (e.g. explanation construction, as introduced in the next sub-section) defined on their parent relation. Thus we need only define the properties and reasoning operations once on the meta-level relations, and all its subclass relations that express much richer domain-specific meanings can inherit them automatically. The other benefit is that new semantic relations can be easily incorporated through defining them as the subclasses of one of the meta-level relations.

3.2 Explanation Construction

Explanation construction is to set up explanation paths between concepts in the semantic network, which are used to explore solutions for particular knowledge-intensive question selection tasks.

We have defined two levels of explanation construction operations. The first level is called “Direct Explanation Construction”, which is suitable when there is a direct (local) relation between two concepts. For example, if there are two questions $Q(A)$ and $Q(B)$ and there is a relation “A is a subclass of B”, then a direct explanation is constructed that “ $Q(A)$ is ranked after $Q(B)$ because A (one possible answer of $Q(A)$) is a lower level concept than B (one possible answer of $Q(B)$)” in the knowledge-intensive question ranking phase.

The second level is referred to as “Transitive Explanation Construction”, which is suitable where there is no direct relation between two concepts in the knowledge base,

but we can set up a new semantic relation between them through exploring other relations in the knowledge base.

The transitive explanation construction is based on the transitivity property defined on different relations. In the meta-level knowledge model, we define the transitivity property on the “infers” relation, the “appears after” relation and the “succeeds” relation, and all their sub-class relations can inherit such property from them. So in each relation category (formed by one of these three basic relations and its sub-class relations), all the subclass relations can be transferred on each other to construct new super-class type relations.

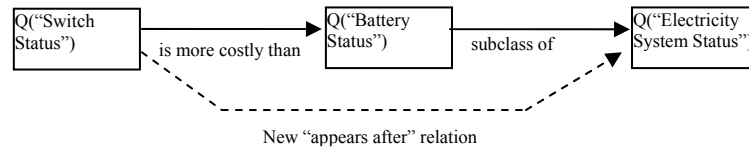


Fig. 5. How to construct a new “appears after” relation

Fig. 5 gives an example of how to build up a new explanation path in the “appears after” relation category through exploring two different subclass relations. In this figure, there are two relations: Q(“Switch Status”) “is more costly than” Q(“Battery Status”) and Q(“Battery Status”) is a “subclass of” Q(“Electricity System Status”). Following the “is more costly than” relation and the “subclass of” relation, a new “appears after” relation, Q(“Switch Status”) “appears after” Q(“Electricity System Status”), is constructed. Thus if we have two questions Q(“Switch Status”) and Q(“Electricity System Status”), we can rank them through constructing the explanation path that “Q(“Switch Status”) should be asked after Q(“Electricity System Status”), because to answer Q(“Switch Status”) is more costly than to answer Q(“Battery Status”), and Q(“Battery Status”) is a lower level question than Q(“Electricity System Status”) in the concept taxonomy about the electricity system fault.

As discussed in the previous subsection, the “joins” relation does not have the property of transitivity. So we can only use the “Direct Explanation Construction” operation to construct explanations to accomplish the consistent question clustering task.

In the CREEK representation, each relation has a default explanation strength attached to it. The explanation strength of a constructed chain of linked relations, which constitute an explanation path, is calculated on the basis of these defaults (in our implementation introduced in Section 4, we will simply use the product of the defaults to indicate the explanation strength of the constructed explanation path).

3.3 Explanation-Boosted Reasoning Process

The explanation-boosted reasoning process can be divided into three steps: ACTIVATE, EXPLAIN and FOCUS. The three steps, which constitute a general process model for knowledge-intensive CBR, was initially described for the retrieve phase [1], although it applies in principle to all four phases of the CBR cycle. Here this model is instantiated for the different question selection tasks. ACTIVATE

determines what knowledge (including case-specific knowledge and general domain knowledge) is involved in one particular task, EXPLAIN builds up explanation paths to explore possible solutions for that task, and FOCUS evaluates the generated explanation paths and identify the best one/ones for that particular task. The operations, done at each step in accomplishing a knowledge-intensive question selection task, are shown in Table 2.

Table 2. Explanation-boosted Reasoning Process in the knowledge-intensive question selection

	Feature Inferencing	Knowledge-intensive Question Ranking	Consistent Question Clustering	Coherent Question Sequencing
ACTIVATE (identify knowledge)	New case features and the related “infers” relations	Discriminative questions and the related “appears after” relations	Sorted questions and the related “joins” relations	Answered questions in the last conversation session and the “succeeds” relations between them and the discriminative questions in current session
EXPLAIN (construct explanation paths)	Feature inferencing explanation paths	Knowledge-intensive question ranking explanation paths	Question clustering explanation paths	Question sequencing explanation paths
FOCUS (evaluate explanation paths and use them to accomplish particular tasks)	The accepted explanations are transformed to new case features	The accepted explanations are combined together with statistical metrics to rank discriminative questions	The accepted explanations are used to group the sorted questions	The accepted explanations are used to re-rank the discriminative question groups

4 System Implementation

We have implemented our proposed approach within the TrollCreek system [2]. TrollCreek is an implementation of CREEK that contains a graphical knowledge model editor and a knowledge-intensive case-based reasoner. Our implementation adds the conversational process with its explanatory mechanism into the retrieve phase.

We are currently exploring two application domains for our CCBR method, car fault detection, and component retrieval for reuse of useful components when developing image processing software [15]. Car fault detection is an example domain adopted in our group for the study of basic knowledge modeling, and representational and reasoning methods, related to particular research directions (e.g. conversational CBR). The knowledge base in this domain incorporates the car fault detection domain knowledge and 29 stored cases. In the graphic window of the knowledge base, we can select an existing case or create a new case to start a knowledge-intensive conversational case retrieve process.

A conversational retrieve process contains one or several conversation sessions (the number of the sessions depends on when the searcher finds her desired case or whether there are still discriminative questions left).

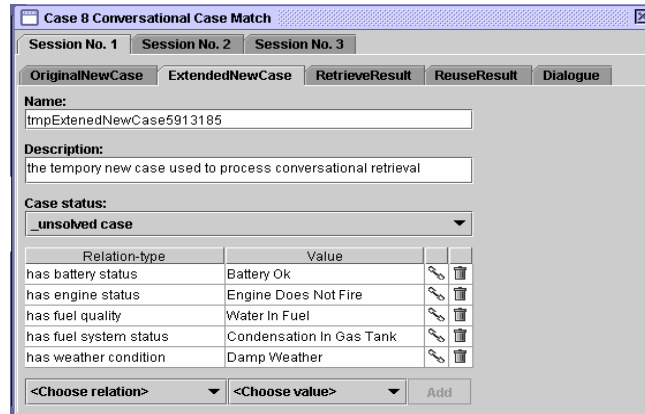


Fig. 6. The “ExtendedNewCase” pane in one conversation session

In the computer interface there are five window panes to move between within each session: The **OriginalNewCase** pane (the example of the content of this pane can be seen in Fig. 2) is used to display the new case in the particular conversation session. The new case extended by the inferred features in the feature inferencing phase is then displayed in the **ExtendedNewCase** pane (as shown in Fig. 6). Based on the extended new case, the CBR retrieve module retrieves a set of sorted cases and displays them in the **RetrieveResult** pane (as illustrated in Fig. 7). In this pane you can inspect the matching details between each retrieved case and the extended new case. The solution for the extended new case is then calculated by the retrieved cases and displayed on the **ReuseResult** pane. If you are not satisfied with the retrieved cases and the reuse result, you can go to the **Dialogue** pane (shown on Fig. 8) to select and answer the discriminative questions, and enter a new conversation session.

The question ranking module divides the identified discriminative questions into two groups: Group one includes the questions that are constrained to be ranked after other questions by some constructed “appears after” explanation paths; Group two contains all the remaining questions. The questions in Group two then gets ranked based on their occurrence frequency metrics [6]. Each question in Group one has one or more “appears after” explanation attached to it. The questions are sorted according to the strongest explanation attached to each questions. Then the ranked questions in Group two are sorted in front of the questions in Group one. If there are some “succeeds” explanation paths between the answered questions in the last conversation session and the current questions, the ranking priority of these involved questions are further increased (putting them in the front of the question queue), and the internal sequence of these “succeeding” questions are decided by their explanation strengths in the “succeeding” explanation paths. The ranked questions are displayed in the Dialogue pane. When each question is selected, its “joined” questions are also displayed in the “Dialogue” pane to prompt the user for further selecting and answering.

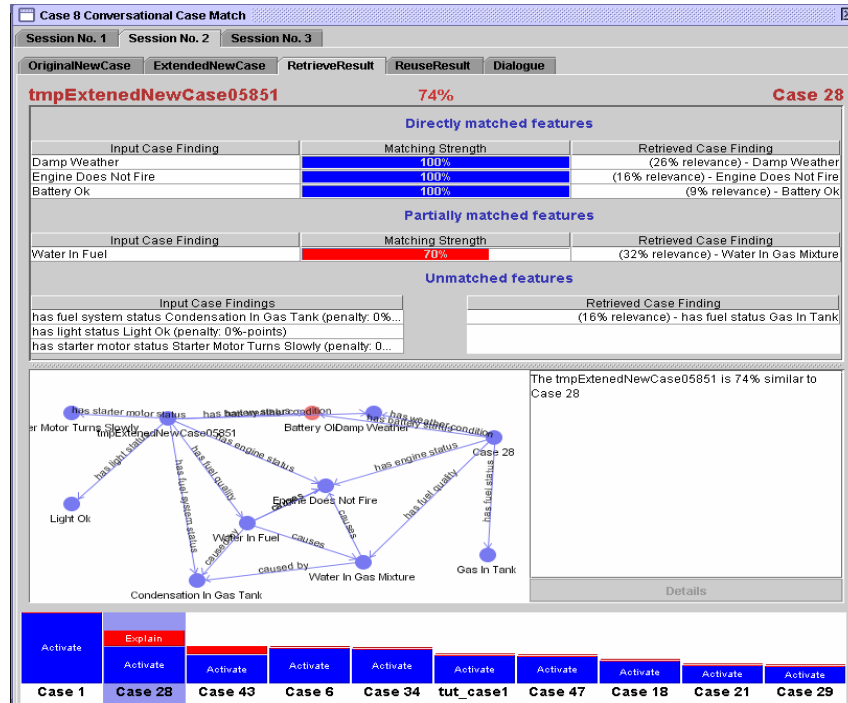


Fig. 7. The “RetrieveResult” pane in one conversation session

Our studies so far indicate that using general domain knowledge as explanatory support in a conversational CBR process improves the focusing of question-asking, and hence reduces the cognitive load needed to identify the best matching case. The target application for empirical testing of our approach will be software component reuse. We are currently building a knowledge base for the components existing in the DynamicImager system [15], a visualization and image processing development environment, in which there are about 200 different image operating components that can be combined in various ways. Our evaluation process will compare component retrieval with and without the explanation method, applied to one-shot vs. conversational CBR retrieval.

5 Related Research

In [22], Schmitt and Bergmann propose a formal model for dialogs between users and a conversation system, in which they identify four important issues in the conversation process: a small number of questions, comprehensible questions, low answering cost of questions and comprehensible question clustering. They also argue that the main reason for the unnatural question sequence during dialogue is due to the ignorance of the relations between different questions. However, they do not give methods about how to incorporate the semantic relations during the dialog process.

In [8], Aha, Maney and Breslow propose a model-based dialogue inferencing (feature inferencing) method. In their method, the general domain knowledge is represented in a library model (including object models and question models) taking the form of a semantic network. At run time, a set of rules are extracted from the library model using an implication rule generator, and the generated rules and the existing problem description are input to a PARKA-DB to infer the implicit knowledge.

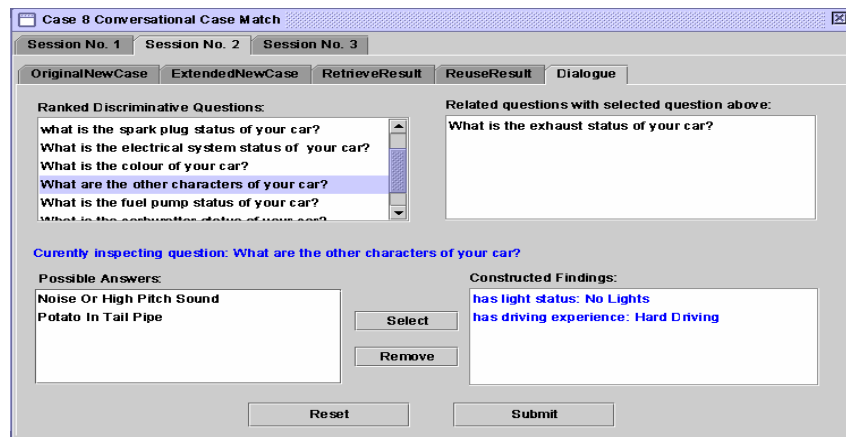


Fig. 8. The “Dialogue” pane in one conversation session

In [17], Gupta proposes a taxonomic conversational CBR approach to tackle the problems caused by the abstraction relations among features. In his approach, cases are described using one or more factors. For each factor, an independent subsumption taxonomy is created by the library designer in advance, and only the most specific feature in each factor taxonomy is selected to describe a case. The similarity between one <question, answer> pair in a case query and one in a case is calculated based on their relative positions in the taxonomy. The question generated from a higher level feature in one factor taxonomy is constrained to be asked before those that come from the lower level features.

Aha, Gupta and Sandhu identify the dependency relation among features [7, 18]. In their method, dependency relations are only permitted to exist between the root nodes among various factor taxonomies and the post-condition node in one dependency relation is excluded from the case representation. In the question ranking step, the question generated from a post-condition node in a dependency relation has higher priority to be asked than the question formalized by the pre-condition node.

Carrick, Yang, Abi-Zeid and Lamontagne try to eliminate the trivial and the repeated questions from users by accessing other information sources to answer them automatically [9]. They take the question answer acquisition costs into account when selecting a task (question) to execute instead of only the information quality metric. In this method, an execution plan is formulated for each question using a hierarchical task network (HTN). The estimated cost for each question is calculated through propagating cost values upward from leaves to the root using the mini-max algorithm.

Comparing with the above knowledge-intensive question selection methods, our approach contributes to the conversational CBR research in two ways: we propose a common integrated framework (including knowledge representation model, explanation construction mechanism and three-step reasoning process) to solve the knowledge-intensive question selection tasks comprehensively (feature inferencing, integrated question ranking, consistent question clustering and coherent question sequencing); and by creating a meta-level knowledge representation model, our approach has the capability to be easily extended to support richer semantic relations that influence the question selection in conversational CBR.

6 Conclusion

The explanation method presented in this paper is based on the CREEK knowledge-intensive CBR approach. The method described extends the existing system with a conversational method and an explanation mechanism targeted at conversational CBR support.

Limitations of the approach include the following two problems. The first is the method's dependence on knowledge engineering. The knowledge base combining both specific cases and general domain knowledge is assumed to exist initially. The construction of this knowledge base puts a significant workload on the development team. However, recent developments in the areas of Knowledge Acquisition and Modeling, as well as Ontology Engineering, provide systematic methods that help reduce this problem [4]. We are also looking into machine learning methods, particularly Bayesian Networks, for solving parts of the problems involved [19].

The second is conflicting knowledge correction. We store the general domain knowledge in the knowledge base, which explicitly expresses the relations among concepts. However, the knowledge provided by users, including the initial problem description and later answers to discriminative questions, can conflict with this stored general domain knowledge. The problem can be reduced by incorporating an automatic mechanism to detect the knowledge conflicts in order to warn users to revise their new cases, or help knowledge base designers to update the predefined mistaken knowledge.

References

1. Aamodt A. Explanation-driven case-based reasoning. *Topics in Case-based reasoning* 1994:274-88.
2. Aamodt A. Knowledge-intensive case-based reasoning in Creek. In Funk P, Calero PAG eds. *7th European Conference on Case-Based Reasoning*. Madrid, Spain: Springer, 2004.
3. Aamodt A. A Knowledge Representation System for Integration of General and Case-Specific Knowledge. *International Conference on Tools with Artificial Intelligence*. New Orleans, 1994:4.
4. Aamodt A. Modeling the knowledge contents of CBR systems. *Workshop Program at the Fourth International Conference on Case-Based Reasoning*. Vancouver: Naval Research Laboratory Technical Note AIC-01-003, 2001:32 - 7.

5. Aamodt A, Plaza E. Case-Based Reasoning: Foundational Issue, Methodological Variations, and System Approaches. *AI Communications* 1994;7:39-59.
6. Aha DW, Breslow LA, Munoz-Avila H. Conversational Case-Based Reasoning. *Applied Intelligence* 2001;14:9-32.
7. Aha DW, Gupta KM. Causal Query Elaboration in Conversational Case-Based Reasoning. International Florida Artificial Intelligence Research Society Conference. Pensacola Beach, Florida, USA, 2002:95-100.
8. Aha DW, Maney T, Breslow L. Supporting Dialogue Inferencing in Conversational Case-Based Reasoning. European Workshop on Case-Based Reasoning. Dublin, Ireland, 1998:262-73.
9. Carrick C, Yang Q, Abi-Zeid I, et al. Activating CBR Systems through Autonomous Information Gathering. International Conference on Case Based Reasoning. Germany, 1999.
10. Cunningham P, Bergmann R, Schmitt S, et al. WEBSSELL: Intelligent Sales Assistants for the World Wide Web. *KI - Kunstliche Intelligenz* 2001;1:28-31.
11. Cunningham P, Smyth B. A Comparison of Model-Based and Incremental Case-Based Approaches to Electronic Fault Diagnosis. Case-Based Reasoning Workshop. Seattle, USA, 1994.
12. Doyle M, Cunningham P. A Dynamic Approach to Reducing Dialog in On-Line Decision Guides. European Workshop on Advances in Case-Based Reasoning. Trento, Italy, 2000:49-60.
13. Göker MH, Thompson CA. Personalized Conversational Case-Based Recommendation. the 5th European Workshop on Case-Based Reasoning (EWCBR 2000). Trento, Italy, 2000.
14. Gu M, Aamodt A. Explanation-boosted question selection in conversational CBR. ECCBR-04 workshop on Explanation in CBR. Madrid, Spain, 2004:105-14.
15. Gu M, Aamodt A, Tong X. Component retrieval using conversational case-based reasoning. In Shi Z ed. International Conference on Intelligent Information Systems. Beijing, China, 2004.
16. Gupta KM. Knowledge-based system for troubleshooting complex equipment. *international Journal of Information and Computing Science* 1998;1:29-41.
17. Gupta KM. Taxonomic Conversational Case-Based Reasoning. International Conference on Case-Based Reasoning. Vancouver, BC, Canada, 2001:219-33.
18. Gupta KM, Aha DW, Sandhu N. Exploiting Taxonomic and Causal Relations in Conversational Case Retrieval. European Conference on Case Based Reasoning. Aberdeen, Scotland, UK, 2002:133-47.
19. Langseth H, Aamodt A, Winnem OM. Learning Retrieval Knowledge from Data. In Anand SS, Aamodt A, Aha DW eds. Workshop ML-05: Automating the Construction of Case-Based Reasoners, in Sixteenth International Joint Conference on Artificial Intelligence. Stockholm, 1999:77 - 82.
20. Mcsherry D. Interactive Case-Based Reasoning in Sequential Diagnosis. *Applied Intelligence* 2001;14:65-76.
21. Schmitt S. simVar: A Similarity-Influenced Question Selection Criterion for e-Sales Dialogs. *Artificial Intelligence Review* 2002;18:195-221.
22. Schmitt S, Bergmann R. A Formal Approach to Dialogs with Online Customers. The 14th Bled Electronic Commerce Conference. Bled, Slovenia, 2001:309-28.
23. Shimazu H. ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. *Artificial Intelligence Review* 2002;18:223 - 44.
24. Sørmo F. Plausible Inheritance: Semantic Network Inference for Case-Based Reasoning. Department of Computer and Information Science. Trondheim: Norwegian University of Science and Technology, 2000:102.