Explanation-Boosted Question Selection in 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. A core research concern in conversational case-based reasoning (CCBR) is how to select the most discriminative and natural questions to ask the user in the conversational process. There are two ways to realize this task: one is to remove the questions whose answers can be inferred from the information a user has provided, which is called dialogue inferencing; the other is to rank the questions to guarantee the most informative questions are asked first, which is referred to as question ranking. In this paper, we present a common explanation-boosted CCBR approach, which utilizes both general domain knowledge and case-specific knowledge to realize dialogue inferencing and question ranking. This approach provides a flexible meta-level knowledge representation model to be able to incorporate richer semantic relations. An application of this approach is illustrated in a car fault detection domain.

1 Introduction

Conversational case-based reasoning (CCBR) [1] is an interactive form of case-based reasoning (CBR)[2]. It uses a mixed-initiative dialog to guide users through a question-answer sequence to refine their problem description incrementally. CCBR applications have been successfully probed in the troubleshooting domain [3], and in the selection of products or services in E-Commerce [4].

As illustrated in Fig. 1, conversational CBR adds user-system interactions to the standard CBR cycle [2]. A user's initial textual problem description is formalized into a structured case query (represented as <question, answer> pairs or <attribute, value> pairs). A CBR retrieve process is executed based on the case query and the knowledge base and a set of retrieved cases, sorted decreasingly by their similarities to the case query, are returned. Unknown questions are identified from the retrieved cases and ranked in a certain way. Both the sorted cases and ranked questions are displayed to the user. The user either find his desired cases, which means the CCBR process is completed, or

```
CaseQuery := Case-Query-formalize(InitialProblemDescription);
Repeat:
    CaseQuery:=Dialogue-Inference(CaseQuery);
    RankedRetrievedCases := CBR-Retrieve(CaseQuery);
    UnknownQuestions := Question-Identify(RankedRetrievedCases, CaseQuery);
    RankedUnknownQuestions := Question-Rank(UnknownQuestions);
    Display(RankedUnknownQuestions, RankedRetrievedCases);
    If (users find their desired cases or have no question to answer) then
        Exit loop;
    Else
        UserAnswer := User-Select-and-Answer-Question();
        CaseQuery :=Case-Query-Update(CaseQuery, UserAnswer);
```

Fig. 1. The conversational case retrieval process in CCBR

select a question to answer, which is followed by a new round of retrieve and conversation based on the updated case query.

A core concern in conversational CBR is how to minimize the cognitive load required by users to retrieve their desired cases [4], which requires to select the most discriminative [1, 5, 6] and natural [7] questions in the dialogue process.

Up to now, several methods, such as the information gain [4, 8, 9], the occurrence frequency metric [1], and the information quality [5], have been proposed to realize question selection. However, all the methods mentioned above are basically knowledge-poor, that is, they only take the superficial statistical information into account. However, general domain knowledge also has a potential to play a positive role in selecting questions. For example, in a car fault detection system, if users have said that the fuel pipe is broken, the system should be able to infer that the fuel transmission system has a problem instead of still asking users "what is the status of the fuel transmission system". Another example is that if the answer of question A is easier or cheaper to obtain than question B's, or if the answer of question B can be inferred from that of question A, question A should be prompted to users before question B. The first example is referred to as dialogue inferencing [6] (see the underlined line in Fig. 1), which concerns inferring the potential knowledge from the current known knowledge, so the questions that can be answered implicitly by the current known knowledge would not be prompted to users. The second one is referred to as a knowledge-intensive question ranking [5, 10-13] (corresponding to the line in bold in Fig. 1), which ranks the candidate questions based on their semantic relations besides their statistical metrics.

In this paper, we present an explanation-"boosted" reasoning approach to support knowledge-intensive question selection, in which general domain knowledge is captured and integrated as explanatory machinery to support dialogue inferencing and knowledge-intensive question ranking in the CCBR process. Here, what we mean by explanation-boosted reasoning is a particular method for constructing explanation paths that explore general domain knowledge for question selection tasks. These explanation paths can also be displayed to users to justify the involved intelligent actions.

The rest of the paper is organized as follows. In Section 2, we identify several semantic relations relevant to question selection. In Section 3, our explanation-boosted question selection approach is introduced. In Section 4, an application of our approach is illustrated in a car fault detection domain. Related research is summarized in Section 5. Discussions and future work are given at the end (Section 6).

2 Semantic Relations Related to Question Selection

As we discussed in Section 1, general domain knowledge is useful for question selection. In this paper, we identify the following relations among concepts, which influence dialogue inferencing and knowledge-intensive question ranking:

Concept abstraction One factor of a case can be described using concepts at different abstraction levels. The lower level a concept belongs to, the more specifically it can describe this factor. The appearance of a lower level concept can be used to infer the existence of its higher concepts. For example, the concept of "fuel transmission mistake" is a lower level concept than that of "fuel system mistake". Here, we define a relation "subclass of" to express the relation of "concept abstraction". "A is a subclass of B" means A is a lower level concept than B. When it comes to question selection, this relation can be used in two ways. In dialogue inferencing, if A is a subclass of B and we have A, then we can infer B (i.e. we need not ask the question about B). In question ranking, if A is a subclass of B then a question about A should be asked after the question about B [11, 12].

Dependency relations We say there is a dependency relation between two concepts if the appearance of one concept 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. Here, we define a relation "depends on" to describe dependency relations. "A depends on B" means B is the necessary condition for A. This relation can also be used in question selection. In dialogue inferencing, if A depends on B and we have A, then we can infer B. In question ranking, if A depends on B, then a question about A should be asked after the question about B [10, 11].

Causality relations The causality relation means one concept can cause the occurrence of another concept. For example, an electricity system mistake in a car can cause its engine not to start. Here, we define a relation "causes" to express the causality relation. "A causes B" means B is the result of A. We can make use of this relation in question selection. In dialogue inferencing, if A causes B and we have A then we can infer B. In question ranking, if A is caused by ("caused by" is the inverse relation of "causes") B then a question about A should be asked after the question about B since if we get B from the question about B, we need not ask users the question about A.

Correlation relations A particular relation, "correlates", is defined to express the relationship between two concepts that they always happen together, even though we can not tell which one causes the other. This "correlates" relation can only be used in dialogue inferencing (from each of these two concepts, we can infer the other), but not in question ranking.

Practical costs The costs to obtain answers to different questions are various. For instance, to test whether a switch has a mistake is more difficult than to test whether the battery has electricity. The relation "more costly than" is defined to represent that to obtain the answer to one question is more difficult than to obtain the answer to another question. This "more costly than" relation can be used in question ranking: if A is more costly than B, then A should be asked after B [5].

3 An Explanation-Boosted Question Selection Approach

In this section, we introduce our explanation-boosted question selection approach from three perspectives: knowledge representation, explanation construction, and explanation-boosted reasoning.

3.1 Knowledge Representation

Knowledge is represented at two levels in our approach: the first one is the object-level, in which case-specific knowledge and general domain knowledge are represented within a single representation framework; the other is the meta-level, which is used to express the inter-relations of the semantic relations introduced in Section 2.

3.1.1 An Object-Level Knowledge Representation Model

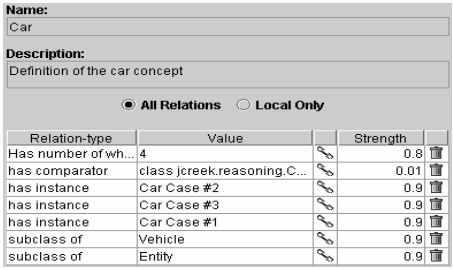

Fig. 2. The frame structure for the concept of car in CREEK

In our approach, a frame-based knowledge representation model, which is a part of the CREEK system [14-16], is adopted to represent the object-level knowledge. In CREEK, both case-specific knowledge and general domain knowledge is represented as concepts, and a concept takes the form of a frame-based structure, which consists of a list of relationships. A relationship is described using an ordered triple $<C_f, T, C_v>$, 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 relationship type. 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 relationship corresponds to a link between two nodes.

Fig. 2 illustrates, in a frame view, how the car concept in the car fault detection domain is represented in CREEK. Fig. 3 shows, in a network view, a part of the knowledge base for that domain. As seen, the semantic relations identified in Section 2 are represented as relationships connecting different concepts. Cases are integrated into the general domain model, since all case features are defined as concepts in it.

The concepts whose instance concepts appear in the retrieved cases, but do not appear in the case query can be converted into discriminative questions. For example, the concept "fuel pumping status" has an instance concept "can pump fuel" appearing in the retrieved cases, but has no instance concepts appearing in the case query, so a discriminative question "what is the fuel pumping status of your car" is added to the discriminative questions list.

In this paper, we define a function, Q: concepts set → questions set, to complete the operation of transforming from a concept into a question. On this function, we further define the following properties:

- The question transformed from one concept is the same as the questions formed by its instance concepts. For example, Q("fuel pumping status")=Q("can pump fuel")=Q("can not pump fuel")= "what is the fuel pumping status of your car".
- A set of concepts that share the same transformed question are referred to as a SQCS (Same Question Concepts Set). We only predefine one question for each SQCS, which is connected with the super-concept within the SQCS.
- The semantic relations that exist between two SQCSs are transferred to the two questions generated from these two SQCSs, for instance, the "depends on" relation that "can pump fuel" depends on "has fuel" is transferred to Q("can pump fuel") depends on Q("has fuel"), that is, the question "what is the fuel pumping status of your car" depends on the question "is there any fuel in you fuel tank".

3.1.2 A Flexible Meta-Level Representation Model and Its Reflective Reasoning Method

We define a basic relation "infers" for dialogue inferencing. The "infers" relation means that if A infers B, we can get B from the existence of A. This relation has the

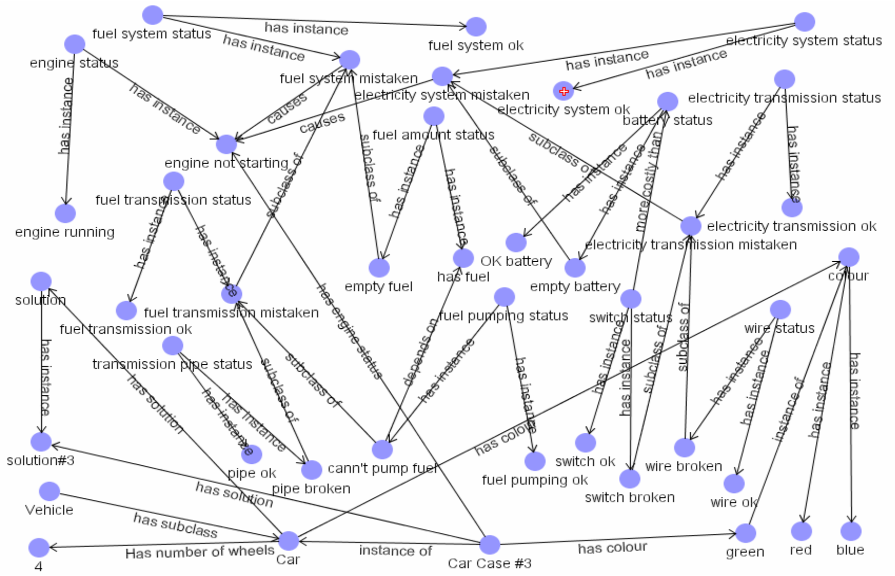

Fig. 3. A part of the knowledge base in the car fault detection application in CREEK

property of transitivity that if A infers B and B infers C then A infers C. We define several semantic relations identified in Section 2, "subclass of", "depends on", "causes" and "correlates" as the subclasses of "infers" since all these relations can be used to infer the existence of a post-condition based on the appearance of the pre-condition.

The other fundamental relation "follows" is defined for question ranking. So, "A follows B" means that Q(A) should be asked after Q(B). This relation also has the property of transitivity that if A follows B and B follows C, then A follows C. We define several relations identified in Section 2, "subclass of", "depends on", "caused by" and "more costly than" as the subclasses of "follows" because all these relations can rank the pre-condition question to be asked after the post-condition question. Fig. 4 illustrates the meta-level structure for semantic relations described above.

One type of reflective reasoning operation, basic subclass inheritance, is made explicit in this meta-level knowledge representation model. Subclass inheritance makes subclass relations inherit the properties and reasoning operations (i.e. the explanation construction introduced in next sub-section) defined on their parent relations. Thus we need only define the properties and reasoning operations once on the parent relations ("infers" and "follows"), and all their subclass relations, which 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 "infers" or "follows", and a

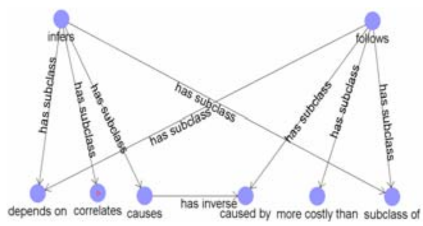

Fig. 4. The semantic relation hierarchy, for reflective reasoning, used to realize dialogue inferencing and question ranking

new application can be easily created through the same process.

3.2 Explanation Construction

Here, explanation construction is setting up a explanation path between two concepts and use it to explore solutions to question selection tasks.

We have defined two levels of explanation construction operations on the "infers" and "follows" relations. The first level is called "Direct Explanation Construction", which is suitable for the case that there is a direct relation between two concepts in the knowledge base. In dialogue inferencing, if concept A exists in a case query and there is a relation "A infers B" in the knowledge base, concept B can be inferred directly and can be integrated into the case query (so Q(B) will be removed from the potential discriminative questions list). In question ranking, if there are two questions Q(A) and Q(B) and there is a relation "A follows B", Q(A) is ranked after Q(B).

The second level is referred to as "Explanation Path Construction", which is suitable when there is no direct relation between two concepts in the knowledge base, but we can set up the "infers" or "follows" relation between them through exploring other relations in the knowledge base. In our group we have developed an abduction-based inference method referred to as plausible inheritance[14, 15], which is adopted to build up the explanation path.

Plausible inheritance is a general relation transitivity mechanism, based on which a relation on one concept can be transferred to another concept following not only the traditional "subclass of" and "instance of" relations, but also other relations, such as "is-part-of", "depends on", "causes" and so on.

In our approach, we define that both the "infers" relation and "follows" relation can be inherited (plausible inheritance) over themselves. So the transitivity property of "infers" relation and "follows" relation is realized. Through combining the subclass inheritance defined on the meta-level knowledge representation model and the plausible inheritance, the "infers" relation and its subclass relations can be transferred over each other. The transitivity property on the "follows" relation and its subclass relations is realized in the same way.

Fig. 5 illustrates an example of how to use plausible inheritance to build up an explanation path for question ranking. In Fig. 5, there are two relations: Q("battery status") is a subclass of Q("electricity system status"), and Q("switch status") is more costly than Q("battery status"), so following the "more costly than" relation, the first relation that Q("battery status") is a subclass of (follows) Q("electricity system status") is transferred to Q("switch status") that Q("switch status") follows Q("electricity system status"). Thus the question ranking explanation path from Q("switch status") to Q("electricity system status") is constructed. Thus if we have two questions Q("switch status") and Q("electricity system status"), we can rank them so that Q("switch status") should be asked after Q("electricity system status") through constructing the above explanation path using plausible inheritance.

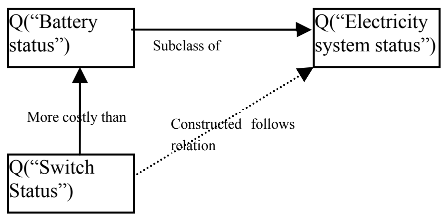

Fig. 5. A "follows" relation is transferred to the concept Q("switch status") using plausible inheritance

In our approach, 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 the example shown in Section 4, we will simply use the product of these defaults to indicate the explanation strength of the constructed explanation path).

3.3 Explanation-Boosted Reasoning

The explanation-boosted reasoning process can be divided into three steps: ACTIVATE, EXPLAIN and FOCUS. ACTIVATE determines what knowledge (including case-specific knowledge and general domain knowledge) is involved in one particular task, EXPLAIN builds up different explanation paths to explore general domain knowledge related solutions for that task, and FOCUS is used to evaluate the generated explanation paths and identify the practical or best one/ones. A similar process is used in the retrieve phase in CREEK to explore the semantic similarities between a case query and stored cases. In this paper, this common explanation-boosted reasoning process is extended to support dialogue inferencing and knowledge-intensive question ranking tasks. The detailed reasoning steps will be illustrated in Section 4 using an example in a car fault detection application.

4 Exemplified Dialogue Inferencing and Question Ranking

In this section, the knowledge representation models, the explanation construction operations and the explanation-boosted reasoning process, introduced in Section 3, are combined together to illustrate how the dialogue inferencing and knowledge-intensive question ranking are completed in a car fault detection application (part of the domain knowledge used in this section can be found in Fig. 3).

4.1 Explanation-Boosted Dialogue Inferencing

In our approach, dialogue inferencing is tackled through using three steps ACTIVATE, EXPLAIN, and FOCUS.

In the ACTIVATE step, all the relevant knowledge with dialogue inferencing is activated, which includes the case query knowledge and semantic dialogue inferencing relations related to this case query knowledge from the knowledge base. For instance, we have the following case query knowledge (CQK):

- *CQK1: Has fuel transmission status = Fuel transmission mistaken*
- *CQK2: Has electricity system status = Electricity system mistaken*

The activated semantic dialogue inferencing relations (SDIR) include:

- *SDIR1: "Fuel transmission mistaken" is a "subclass of" "fuel system mistaken" (weight: 0.9)*
- *SDIR2: "Fuel system mistaken" "causes" "engine not starting" (weight: 0.8)*
- *SDIR3: "Electricity system mistaken" "causes" "engine not starting" (weight: 0.8)*

The EXPLAIN step uses the case query knowledge and activated semantic dialogue inferencing relations to reason or explain what knowledge can be inferred through using explanation construction operations. In this example, we get the following new case query knowledge (NCQK):

- *NCQK1: Has fuel system status = Fuel system mistaken (based on CQK1 and SDIR1, weight: 0.9)*
- *NCQK2: Has engine status = Engine not starting (based on SDIR1, SDIR2, and CQK1, weight: 0.72)*
- *NCQK3: Has engine status = Engine not starting (based on CQK2 and SDIR3, weight: 0.8)*

In the FOCUS step, all the inferred knowledge is evaluated, and the accepted knowledge is combined together with the current case query to form a new case query. In the evaluation process, only the knowledge whose weights surpass a particular threshold (say 0.8) is accepted, and the redundantly inferred knowledge is removed (the knowledge with the highest weight is kept). In our example, we get a new updated case query that includes CQK1, CQK2, NCQK1 and NCQK3.

4.2 Explanation-Boosted Question Ranking

Based on the updated case query, a retrieve process [16] is executed and the top ranked cases are returned. In this stage, if users can not find their desired cases, an explanation-boosted question ranking process is started, which is also divided into three steps.

In the ACTIVATE step, the unanswered questions are identified (see Section 3.1.1). All the semantic question ranking relations concerned with these identified questions are then activated from the knowledge base.

For instance, from the retrieved cases and the updated case query, we identify the following unanswered questions (UQ) (we assume that retrieved cases include all the value concepts appearing in Fig.3):

- *UQ1: Q("transmission pipe status")*
- *UQ2: Q("fuel amount status")*
- *UQ3: Q("fuel pumping status")*
- *UQ4: Q("switch status")*
- *UQ5: Q("battery status")*
- *UQ6: Q("electricity transmission status")*
- *UQ7: Q("wire status")*
- *UQ8: Q("colour")*

The activated semantic question ranking relations (SQRR) include:

- *SQRR1: Q("Fuel pumping status") "depends on" Q("fuel amount status") (weight: 0.8)*
- *SQRR2: Q("Wire status") is a "subclass of" Q("electricity transmission status") (weight: 0.9)*
- *SQRR3: Q("switch status") is "more costly than" testing Q("battery status") (weight: 0.75)*

The EXPLAIN step uses the identified unanswered questions and the activated semantic question ranking relations to reason or explain which questions should be asked before another one. For instance, we get the following question ranking knowledge (QRK) through using explanation construction operations:

- *QRK1: Q("fuel amount status") should be asked before Q("transmission pumping status") (based on UQ2, UQ3, and SQRR1, weight: 0.8)*
- *QRK2: Q("electricity transmission status") should be asked before Q("wire status") (based on UQ6, UQ7, and SQRR2, weight: 0.9)*
- *QRK3: Q("battery status") should be asked before Q("switch status") (based on UQ5, UQ4, and SQRR3, weight: 0.75)*

In the FOCUS step, the semantic ranking knowledge obtained in the EXPLAIN step is evaluated, and the questions are ranked combining the semantic question ranking knowledge and the superficial statistical metrics. In the evaluation process, only the ranking knowledge whose weights surpass one particular threshold (say 0.8) is accepted. In this case, the QRK3 is refused because its explanation strength is less than 0.8. In the question ranking process, all the questions are classified into two groups firstly: group one includes the questions whose ranking priorities are constrained by the question ranking knowledge (here, it has two questions: UQ3 (constrained by QRK1), and UQ7 (constrained by QRK2)); and group two contains all the remaining questions (UQ1, UQ2, UQ4, UQ5, UQ6 and UQ8). Secondly, the questions in group two are further ranked based on their superficial statistical metrics such as information gain or occurrence frequency. The questions in group one are sorted according to their biggest explanation strength selected from all the question ranking explanation strengths each question gets increasingly. In this example, the questions in group one are ordered as UQ3, UQ7. And the ranked questions in group two are prompted to users followed by the sorted questions in group one.

5 Related Research

In [6], Aha, Maney and Breslow propose a model-based dialogue 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 potential knowledge.

In [5], the authors try to eliminate the trivial and the repeated questions from users by accessing other information sources to answer them automatically. They take the cost factor 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.

In [12], 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. On each factor, an independent subsumption taxonomy is created by the library designer in advance, and only the most specific feature on 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 [10, 11]. In their method, dependency relations are only permitted to exist between the root nodes among various factor taxonomies, and the precedent node in one dependency relation is excluded from the case representation. In the question ranking step, the question generated from a precedent node in a dependency relation has higher priority to be asked than the question formalized by the dependent node.

Comparing with the above knowledge-intensive question selection methods, our model contributes to the conversational CBR research in three ways: it provides a common explanation-boosted reasoning process to support both dialogue inferencing and knowledge-intensive question ranking; it can rank discriminative questions through combining both their semantic question ranking relations and their superficial statistical metrics; by creating a meta-level knowledge representation model, our model has the capability to be easily extended to support richer inferring or ranking relations, and to be transformed to other application domains.

6 Discussion and Future Work

We will here address two potential limitations in our approach that need to be tackled in our future work. One is conflicting knowledge correction. We store the general domain knowledge in the knowledge base, which explicitly express the relations among concepts. However, the knowledge provided by users, expressed in case queries, can be conflicting with this stored general domain knowledge. In this case, there should be 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. Another problem is the preference cycle generated by a set of question ranking relations. For example, there are three questions, A, B, and C, and three question ranking relations that A should be asked before B, B should be asked before C, and C should be asked before A, so a preference cycle appears following A, B, C, and A. An automatic preference cycle detecting mechanism in the knowledge input phase will be helpful. Another possible solution is directly ignoring the ranking relation with the least explanation strength in any preference cycle.

References

1. D. W. Aha, L. A. Breslow, and H. Muňoz-Avila, "Conversational Case-Based Reasoning," Applied Intelligence, vol. 14, pp. 9-32, 2001.
2. A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issue, Methodological Variations, and System Approaches," AI Communications, vol. 7, pp. 39-59, 1994.
3. K. M. Gupta, "Knowledge-based system for troubleshooting complex equipment," International Journal of Information and Computing Science, vol. 1, pp. 29-41, 1998.
4. H. Shimazu, "ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops," Artificial Intelligence Review, vol. 18, pp. 223 - 244, 2002.
5. C. Carrick, Q. Yang, I. Abi-Zeid, and L. Lamontagne, "Activating CBR Systems through Autonomous Information Gathering," International Conference on Case Based Reasoning, Germany, 1999.
6. D. W. Aha, T. Maney, and L. Breslow, "Supporting Dialogue Inferencing in Conversational Case-Based Reasoning," European Workshop on Case-Based Reasoning, Dublin, 1998.
7. M. Doyle and P. Cunningham, "A Dynamic Approach to Reducing Dialog in On-Line Decision Guides," European Workshop on Advances in Case-Based Reasoning, Trento, 2000.
8. M. H. Göker and C. A. Thompson, "Personalized Conversational Case-Based Recommendation," European Workshop on Case-Based Reasoning, Trento, Italy, 2000.
9. H. Shimazu, A. Shibata, and K. Nihei, "ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops," Applied Intelligence, vol. 14, pp. 33-48, 2001.
10. D. W. Aha and K. M. Gupta, "Causal Query Elaboration in Conversational Case-Based Reasoning," International Florida Artificial Intelligence Research Society Conference, Pensacola Beach, Florida, USA, 2002.
11. K. M. Gupta, D. W. Aha, and N. Sandhu, "Exploiting Taxonomic and Causal Relations in Conversational Case Retrieval," European Conference on Case Based Reasoning, Aberdeen, Scotland, UK, 2002.
12. K. M. Gupta, "Taxonomic Conversational Case-Based Reasoning," International Conference on Case-Based Reasoning, Vancouver, BC, Canada, 2001.
13. D. Mcsherry, "Interactive Case-Based Reasoning in Sequential Diagnosis," Applied Intelligence, vol. 14, pp. 65-76, 2001.
14. F. Sørmo, "Plausible Inheritance: Semantic Network Inference for Case-Based Reasoning," Master thesis in Department of Computer and Information Science. Trondheim: Norwegian University of Science and Technology, pp. 102, 2000.
15. A. Aamodt, "A Knowledge Representation System for Integration of General and Case-Specific Knowledge," International Conference on Tools with Artificial Intelligence, New Orleans, 1994.
16. A. Aamodt, "Explanation-driven case-based reasoning," Topics in Case-based reasoning, Springer, pp 274-288, 1994.