The Application of Case Based Reasoning on Q&A System

Peng Han, Rui-Min Shen, Fan Yang, and Qiang Yang

Dept. of Computer Science and Engineering, Shanghai Jiao tong Univ., Shanghai, China
{phan, rmshen, fyang } @mail.sjtu.edu.cn

Abstract. Q&A (Question and Answer) system is an important aiding tool for people to obtain knowledge and information from the Internet. In this paper, we introduce CBR (Case Based Reasoning) into traditional Q&A system to increase the efficiency and accuracy of retrieving the solution. We put forward an interactive and introspective Q&A engine which uses keywords of the question to trigger the case and sorts the results by the relationship. The engine can also modify the weights of the keywords dynamically based on the feedbacks of the user. Inside the engine, we use a feature-weight maintenance algorithm to increase the accuracy. We also extend the 2-layer architecture of CBR to a 3-layer structure to make the system more scalable and maintainable.

1 Introduction

Question and Answer (Q&A) System is one of the most important components in E-Learning environment which aims at answering the questions asked by the student during their study processes. Accuracy and efficiency are the main two criteria used to evaluate the Q&A systems. Many Q&A systems have already developed based on email-solution, keyword-matching or word-segmentation techniques [12, 13, 14, 15]. With the growth of the number of users and questions, the process time of these systems will become longer and the matching accuracy will become lower due to different presentations of the question and variable interests of the user.

In order to overcome the above disadvantages, we introduce CBR (Case Based Reasoning) into traditional Q&A system. CBR, representing a new generation of expert system technology, has enjoy tremendous success as a technology for solving problems related to knowledge reuse. In this paper, we put forward an interactive Q&A engine based on CBR. This engine uses keywords of the question to trigger case and sorts the results by the relationship and can modify the weights of the keywords dynamically depending on the feedbacks from the user. We also present a new feature-weight maintenance algorithm to increase the accuracy. At last we extend the 2-layer architecture of CBR to a 3-layer structure to make the system more scalable and maintainable.

In next section we present the architecture of the introspective Q&A system based on CBR technique. Section 3 introduces the Case Authoring Module in the architecture and the construction of Case Base in details. We also discuss the definition of question-answer case and the construction of 3-layer Case Base structure

R.I. McKay and J. Slaney (Eds.): AI 2002, LNAI 2557, pp. 704–713, 2002.
© Springer-Verlag Berlin Heidelberg 2002

there. We still introduce the related feature-weight maintenance algorithm. In Section 4 we discuss the experiment results for evaluating the performance of our system. We give our conclusion in Section 5, where we will also explore our future work.

2 Architecture of the Auto Q&A System

Our Q&A system is based on the CBR technology. The whole system is divided into two separate modules, with the first one called Case Authoring Module and the second one introspective Q&A Engine.

The Case Authoring Module is to represent the unstructured field knowledge structurally based on empirical expert knowledge and application background. All these structural representations can be transferred into question-answer instances and stored in the system case base. The introspective Q&A Engine is the kernel of our system. It is triggered by the keywords or description of the problems and returns the similar problems related to the description ranked by the score. So the user can select the most similar problems and get the answer. Furthermore, the system provides a feedback module to adjust the weights of keywords according to the user's score. The architecture of the Q&A system is shown as Figure 1:

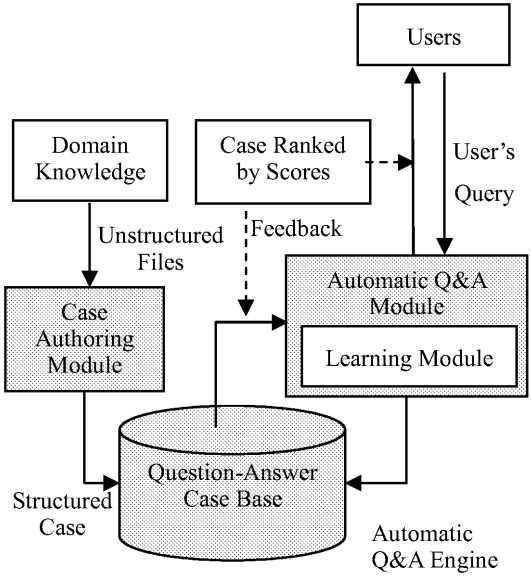

Fig. 1. Architecture of the Q&A System

The system has been used in the professional e-learning site of Shanghai Jiao Tong University (www.nec.sjtu.edu.cn). So all the questions, answers and the relativity between them are accessed through the standard web interfaces. The users, especially the students, produced a great number of questions and potential answers during the learning process. All the questions and answers are assembled in log files.

So we can train the index architecture of the relationship between questions and answers based on the log files. This process is running during the life cycle of the system, which makes the Q&A system become a closed-loop system.

3 Case Authoring Module and Case Base Construction

3.1 Definition of the Question-Answer Case

The description and definition of case is the foundation of a CBR system, and there isn't a uniform standard for it so far since it has strong domain characteristic [7, 8]. In the Case Authoring Module, we define our case-description based on the e-Learning domain characteristics and organize the unstructured domain knowledge in a structural way. The cases in the Q&A system are the description of question and answer. The representation of a case is as follows:

Keywords: short description of the case, which can be used in fuzzy string matching with the user's initial free-form text input.

Attributes: the features that present the main content and characteristics of the question.

Question Description: This is a more detailed textual description of the question's object or content used to confirm the general problem area.

Answer: The answer provides a solution to the case in either textual format or any multimedia format.

Table 1 and Table 2 give an example of the case representation for the Q&A system. The cases in Table 1 are fairly refined, down to the detailed features and their values, while the cases in Table 2 have only two major parts: problem description and answer.

Table 1. Case Representation with Detailed Features (Attributes are not shown due to lack of space)

Type	Keyword1	Keyword2	Answer
Concept & Difference	Switcher	Router	Describe the concepts and difference of Switcher and Router.

Table 2. Case Representation with Natural Language Description

No.	Problem Description	Answer
1	What's the difference between the Switcher and Router?	The function of switcher is quite different from the router …… (Just present the difference between them)
2	What are the concepts of Switcher and Router?	Switcher is the …… Router is the …… (Just list the concepts of them)

3.2 The 3-Layer Architecture of the Question-Answer Case Base

After the description of the case has been defined, the next essential task is to construct the case base and feature index. Each case is associated with a set of feature-value pairs. These pairs are combinations of important descriptors of a case, which distinguish it from other cases. So a case base could naturally be viewed as 2-layer architecture comprised of the feature-value layer and case layer. Using the weights assigned to the connections between the feature-value pairs and the case, a CBR system determines the most relevant cases ranked by the feature information submitted by the user and then returned the results to the user for considerations.

However, in practical implementations we find that the definition of the 2-Layer structure sometimes does not work well since when the number of cases becomes too large the efficiency and scalability of the system will decrease dramatically. So we expand the 2-layer architecture into a 3-layer by dividing the case layer into question layer and answer layer. Additionally, we introduce a second set of weights, which attaches to the connections between questions and their possible answers. This second set of weights represents how important an answer is to a particular question.

In order to describe the situation of user's query, we extend the traditional 2-level case base architecture into a 3-level network structure and take the 2-layer structure as a special case. Consider each case as presented as a triple <F, P, S>, where F corresponds to the feature values, P are the problem description and S are the answers. We can split this representation into three levels: a feature level corresponding to feature values F, a problem description level corresponding to P and an answer level corresponding to S. With this model, when a user enters the feature-values at the first level, the system ranks problem descriptions for users' consideration at the middle layer. The user then selects one intended problem description and the system provide the possible answers which is also ranked according to the second set of weights. New system architecture is shown in Figure 2.

An important motivation for this separation in a case is to reduce the redundancy. Given N cases and M solutions, a case base of size $N \times M$ is now reduced to the one of size $N + M$, which eases the scale-up problem and helps make the case base maintenance task easier. A solution can now be shared by several cases and will only need to be revised once if needed.

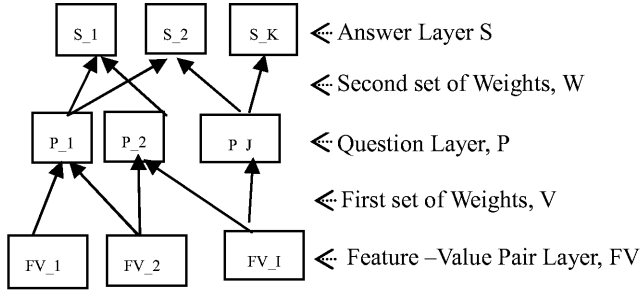

Fig. 2. Tree-layer Architecture of a Question-Answer Case Base

4 The Rank Computation and Maintenance Algorithm

There are two sets of weights, which is similar to the weights in a 3-layer back-propagation neural network. Suppose that there are N features. For each feature F_i, there are m_i values, where i=1, 2… N. The case base contains J problems and K answers. For the architecture shown in Figure 2, there is a total of I= $\sum_{i=1}^{N} m_i$ feature-value pairs, that is to say, there are I nodes in the feature-value layer. We label these feature-value pairs as FV_i, i=1,2,…,I. In the problem layer, we use P_j to represent each problem, where j=1,2,..,J. In the answer layer, we use S_k to represent each answer, where k=1,2,…,K.

The first set of weights $V_{j,i}$ is attached to the connection between a problem P_j and a feature-value pair FV_i if there is an association between them. The second set of weights $W_{k,j}$ is attached to the connection between a answer S_k and a problem P_j if S_k is an answer to P_j.

Given the feature-value pairs selected by a user, the corresponding nodes at the feature-value layer are turned on. A problem's score is computed base on those selected feature-value pairs. For each problem P_j, its score is computed using the following formula:

$$S_{P_j} = \frac{2}{1+e^{-\lambda * \sum_{i=1}^{I}(V_{j,i} * X_i)}} - 1 \tag{1}$$

where $j = 1,2,\cdots,J$, S_{Pj} is the score of the problem P_j, and X_i is 1 if there is a connection between problem P_j and feature-value pair FV_i , then FV_i is selected. Otherwise X_i is 0.

After the problem scores are computed, the problems and their scores will be presented to the user for selection and confirmation. For the current *selected confirmed* problem, the computation of an answer's score is also similar to the computation of an output in a back-propagation neural network:

$$S_{S_k} = \frac{2}{1+e^{-\lambda * \sum_{i=1}^{I}(W_{k,j} * S_{P_j} * \alpha)}} - 1 \tag{2}$$

where S_{Sk} is the score of answer S_k, and S_{Pj} is the score of problem P_j. If there is no connection between answer S_k and P_j, then we do not include it in $\sum_{j=1}^{J}(W_{k,j} * S_{p_j} * \alpha)$.

Since the user should first decide which problem at the problem layer is the most desired one based on his or her current preference and this information needs to be reflected in the subsequent computation of the solution score, we introduce a new parameter α into our learning network and call it the *bias factor*. We expected the selected problem to have a higher bias factor than the unselected ones so as to contribute more in the final answer scores. Thus the answers of the selected problems might have relatively higher scores.

The computation of the learning delta value is first done at the answer layer [16, 17]. We only compute the delta values for the answers associated with the current *selected and confirmed* problem. The following formula is employed:

$$\delta_{S_k} = \frac{1}{2} * (D_{S_k} - S_{S_k}) * (1 - S_{S_k}^2) \tag{3}$$

where δ_{Sk} is the learning delta value for answer S_k, and D_{Sk} is the desired score for S_k.

The learning delta values are then propagated back to the problem layer. The computation of the delta value at this layer is done using the following formula:

$$\delta_{P_j} = \frac{1}{2} * (1 - S_{P_j}^2) * \sum_{k=1}^{K} (\delta_{S_k} * W_{k,j}) \tag{4}$$

where δ_{pj} is the learning delta value of problem P_j. If there is no connection between answer S_k and problem P_j, we do not include it in $\sum_{k=1}^{K} (\delta_{S_k} * W_{k,j})$.

After computing S_{Pj}, S_{Sk}, δ_{Pj}, δ_{Sk}, we need to adjust the weights of the connections between the answer layer and the problem layer, and then connections between the problem layer and the feature-value pair layer. We will adjust the weights attached to the answers which are associated with the current *selected and confirmed* problem. The formula for this adjustment is:

$$W_{k,j}^{new} = W_{k,j}^{old} + \eta * \delta_{S_k} * S_{Pj} \tag{5}$$

where $W_{k,j}^{new}$ is the old weight to be computed, and $W_{k,j}^{old}$ is the new weight attached to the connection between answer S_k and P_j.

The weights attached to connections between the problems and the feature-value pairs will be adjusted next using the learning delta values as follows:

$$V_{j,i}^{new} = V_{j,i}^{old} + \eta * \delta_{P_j} * X_i \tag{6}$$

where $V_{j,i}^{new}$ is the new weight to be computed, $V_{j,i}^{old}$ is the old weight attached to the connection between problem P_j and feature-value pair FV_i. X_i is 1 if there is a connection between them and FV_i is selected by the user. Otherwise X_i is 0.

In addition to scaling-up and redundancy advantages, an added advantage of this architecture is that we can now represent a context sensitive case base. In this way, the second layer, which consists of problem descriptions, can be used to represent both the problem and the context layers, the latter representing different contexts in which problems occur. Under such conceptual representation, the third layer now contains the actual cases. A user can enter a problem's description in the form of feature value pairs and then select the desired context in which to solve the problem. The second set weights in turn can help rank the right case for solving the problem. A set of features can simultaneously influence the contexts and the cases at the same time.

5 Case Study and Experiment

We have implemented the introspective Q&A system based on our professional distance learning web sites. Figure 3 to Figure 5 show an example of the process to solve a problem (since it is a system for Chinese users, so even the English version still has some Chinese information). As can be seen in Figure 3, user can first enter a problem description to identify the context under which the problem is solved. User can describe the problem with any natural description language as he wishes.

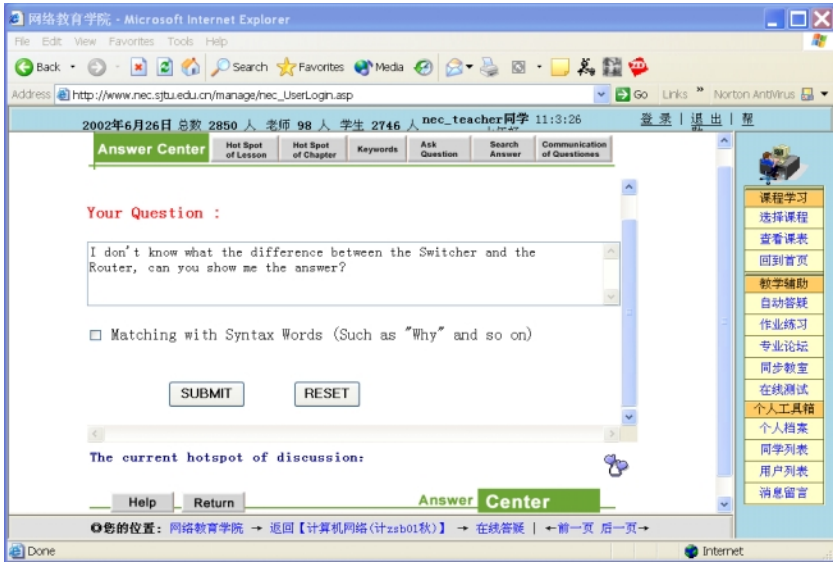

Fig. 3. Question Submission in Q&A System

After submitting the question, a collection of initial cases which match the partial description are retrieved and returned to the user. These cases serve as the candidate answers for subsequent problem solving. Questions that are associated with the candidate cases are then presented to the user (as shown in Figure 4).

The system will return the questions ranked by the average score. Then user can select any similar question to see the answer. If they think the ranking and the score of the question is not fit for him, he can adjust the score in the answer showing column. And the system will then adjust the corresponding weights based on the score given by the user and use the new weights to calculate score next time (As shown in Figure 5).

In order to evaluate the efficiency and validity of our algorithm, we give the test results based on the Network Course database from the NEC Question/Answer Repository. The Network Course database contains 300 instances and 28 attributes. We divide the main case base into several incremental sub-case base, containing 50, 100, 150, 200, 250 instances respectively.

In our experiment, we first convert these databases into the case bases that our algorithm can handle by converting all rows into cases and all columns into features

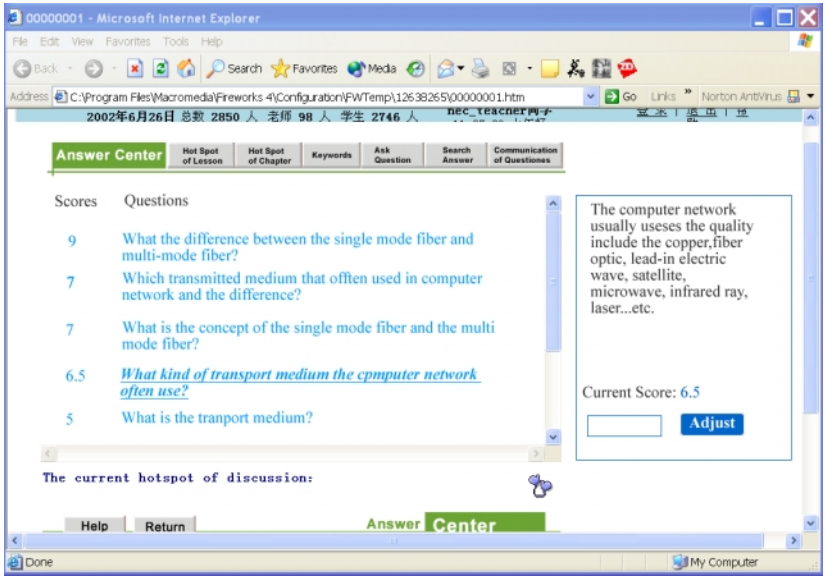

Fig. 4. The Returned Similar Question Lists Ranked by Average Scores

In these tests, the score of a case or an answer is between 0.0 and 1.0; meanwhile, suppose the initial values of the weight are 0.5.

Based on the 3-layer architecture, we perform ten training based on the sub-case bases. Finally, we can get the mean-errors convergence plot (As shown in Figure 6).

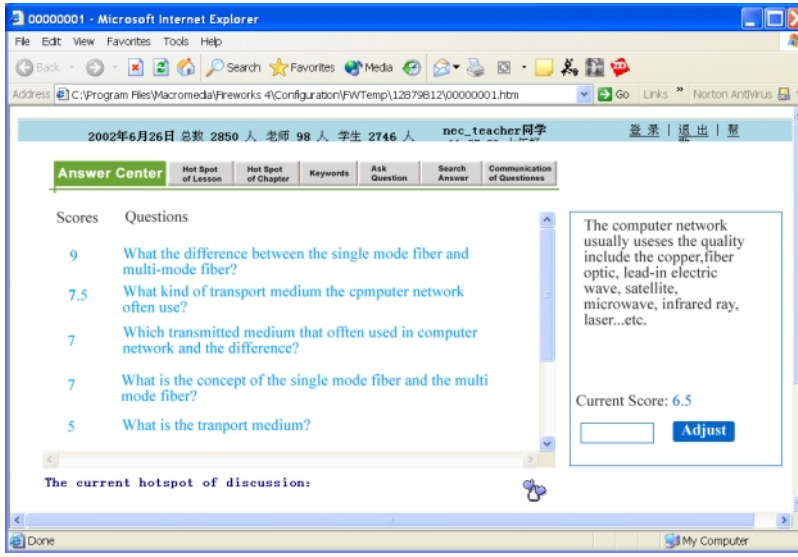

Fig. 5. The Adjusted Question List Based On the Learning Network

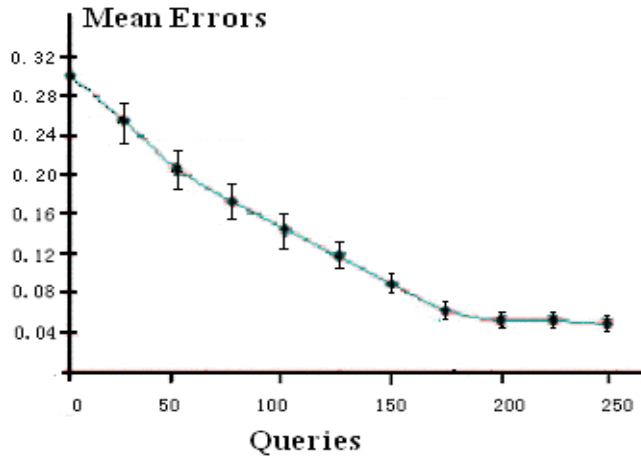

Fig. 6. Plot of the mean errors convergence and 95% confidence interval along the CBR process

In this figure, the 95% confidence interval is also shown on each datum point, where the size of the interval indicates the fluctuation around the mean values, and the average processing time is approximately 1.8 seconds.

6 Conclusion and Future Work

Our work aims at achieving the goal of implementing a Question and Answer system with high efficiency and accuracy to help the users in E-Learning get the professional direction in time. The 3-layer Q&A system based on Case Based Reasoning technique can improve the scaling-up and redundancy of the Q&A system. Furthermore, the context sensitive case base due to the 3-layer architecture can also make the ranking more accuracy. The test based on the real professional site database also proves the performance of our algorithm.

The system has a number of areas to be improved. With the continual growth of the user and the expert fields, the number of cases may become very large, and as a result, the problem will become more complex and the number of returned similar cases will become extra large. In our future work, we will try to find an effective cluster method to merge the similar cases together. And find a way to reduce the number of returned similar cases.

References

1. D. B. Leake. CBR in context: The present and future. In David B. Leake, editor, Case-Based Reasoning, Experinces, Lessons & Future Directons, page 1–30. AAAI Press/ The MIT Press, Menlo Park CA, USA, 1996.

2. I. Watson. Appling Case-Based Reasoning: Techniques for Enterprise System. Morgan Kaufmann Publishers, Inc.,1997.
3. Costas Tsatsoulis, Qing Cheng, Hsin-Yen Wei . Integrating Case-Based Reasoning and Decision Theory. IEEE Transactions on Intelligent Systems, Vol. 12, No. 4, pp. 46–55, 1997.
4. D. B. Leake, A. Kinley, and D.Wilson. Learning to improve case adaptation by introspective reasoning and CBR. In Proceedings of the First International conference on Case-Based Reasoning, pages 229–240, Sesimbra, Portugal, 1995. ISO Publishers.
5. Masaaki Takahashi, Jun-ichi Oono, Kazuyuki Saitoh, Shunji Matsumoto. Reusing Makes It Easier: Manufacturing Process Design by CBR with KnowledgeWare, IEEE Transactions on Intelligent Systems, Vol. 10, No. 6, pp. 74–80, 1995.
6. Ivo Vollrath, Wolfgang Wilke, Ralph Bergmann. Case-Based Reasoning Support for Online Catalog Sales, IEEE Transactions on Internet Computing. Vol. 2, No. 4, pp. 47–54, 1998.
7. K.Racine and Q.Yang. Maintaining Unstructured case bases. In proceedings of the Second International Conference on Case-Based Reasoning, ICCBR-97, pages 553–564, Provindence RI, USA, 1997.
8. Barry Smyth, Mark T. Keane, Pádraig Cunningham. Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving Techniques for Plant-Control Software Design. IEEE Transactions on Knowledge and Data Engineering. Vol. 13, No. 5, pp. 793–812, 2001.
9. V. Ganti, J. Gehrke, R. Ramakrishnan. Mining very large databases. COMPUTER, 32(8):38–45, 1999.
10. Zhong Zhang and Qiang Yang. Feature Weight Maintenance in Case Bases Using Introspective Learning. Journal of Intelligent Information Systems, Kluwer Academic Publishers, 16, Pages 95–116, 2001. The Netherlands.
11. Qiang Yang and Jing Wu. Enhancing the Effectiveness of Interactive Case-Based Reasoning with Clustering and Decision Forests. In Applied Intelligence Journal; Special Issue on Interactive CBR (Editors: David Aha and Hector Munoz-Avil). Jan/Feb 2001. Vol 14, No. 1. Pages 49–64. Kluwer Academic Pubslishers.
12. http://www.ejiajia.com
13. http://www.sijiehe.com
14. http://www.ibm.com/FAQs
15. http://www.mit.ed
16. D.E.Rumelhart and J.L.McClelland，editors. Parallel Distributed Processing, MIT Press, Cambridge, Massachusetts, 1986.
17. G.. Hinton and J.Anderson. Parallel Models of Associative Memory. Lawrence Erlbaum, Potomac, Maryland, 1981.