Learning Assistance Mechanism Using Case-Based Reasoning

Ban Kaku, Tetsuya Matsumoto and Norimichi Kojo
Software Research Laboratory, Sapporo Division, Japan

Guo Xin
Shenyang Administration College, China

SUMMARY

An intelligent learning system can provide users with different learning materials and questions of fixed difficulty according to the learner’s skill and knowledge. However, even the same question might have a different difficulty for different learners or at different learning stages. In this paper, we propose a learning assistance mechanism composed of the following functions: a question-setting assistance function that involves dynamically evaluating the difficulty of the question based on the knowledge state of the learner, and an answering assistance function that involves predicting the correctness of the answer, using case-based reasoning. In this mechanism, the knowledge state, treated as case components for each question, is extracted from the learner’s knowledge model, and a case-base search is performed to find the correctness probability from analogous cases. All non-learned questions are dynamically arranged in the order of high correctness probability. Questions with high correctness probability have low difficulty, and conversely, those with low correctness probability have high difficulty. The teaching strategy for selecting a suitable question in accordance with the knowledge state of the learner is discussed. The mechanism collects a database of right answer cases and wrong answer cases from the answer history of the learner, for use in predicting his or her correctness to the current question. Based on the prediction result, the learner is offered assistance messages and a sorted keyword list to lead him or her towards successful learning. In case-base searching, an evaluation algorithm involving case reliability and relevant keyword importance is proposed to enhance reasoning accuracy. © 1998 Scripta Technica, Syst Comp Jpn, 29(5): 73–83, 1998

Key words: Question-driven learning; case-based reasoning; learning assistance; learner knowledge model, dynamic difficulty evaluation; answer prediction.

1. Introduction

An intelligent learning system is intended to offer an optimized learning environment appropriate to the knowledge state and recognition skills of the learner. However, it is difficult to create reasoning rules for implementing learning assistance adapted to different learner knowledge states associated with learning progress, different learning tasks, and so on. We have discussed various ideas for learning environments in earlier works, such as generating assistance information according to the recognition characteristics of the learner [2] in a question-driven learning system [1], a decision mechanism for a learning-assistance strategy using neural networks [3], and so on. In the current work, we propose a learning-assistance mechanism composed of the following functions: a question-setting assistance function that involves dynamically evaluating the
difficulty of the question based on the knowledge state of the learner, and an answering-assistance function that involves predicting the answer correctness by using case-based reasoning [8]. The combination of the learner’s knowledge state (keyword understanding levels) for each question with the answer correctness information is treated as case data, and the case-base reasoning framework is used for implementing the general reasoning procedure of learning assistance. In addition, each case is created based on analogous learning data rather than being treated as an individual one. Using such case definitions allows a reduction of case-related noise and the enhancement of case reliability.

Generally, the teaching strategy of a learning system offers learning material and questions with a previously assigned difficulty in accordance with the knowledge state and skills of the learner. However, the difficulty of a question might be different for different learners or at different learning stages, even for the same learner. For instance, the same question might appear difficult to learner A, but easy to learner B. Therefore, question-setting control will be more adequate if the teaching strategy uses a question difficulty that is dynamically evaluated based on the current learner’s knowledge state at the current learning stage. In addition, the correctness predicted before the actual answer might be useful for executing a strategy of offering assistance information or for finding hints to improve the learning efficiency [4].

In this paper, the approach of the proposed learning assistance mechanism is explained first, and then the question-setting assistance function and the answering assistance function are discussed separately. After defining the learner’s knowledge model, the case data structure, and the importance of the relevant knowledge keywords, the implementation of each assistance function is described in more detail. A case base search algorithm that takes account of the predicted success ratio (reliability) and the importance of relevant knowledge keywords is proposed. We also discuss issues of the updating of the learner’s knowledge model and the self-organization of the case base.

2. Basis of the Learning Assistance Mechanism

Prior to discussing the basis of the proposed learning assistance mechanism, we will briefly describe the general purpose of the question-driven learning system previously proposed by the authors. The purpose of such a system is to allow the learner to acquire knowledge through searching (referring to) the relevant keywords necessary for answering the question. Currently, the system incorporates 111 questions and a vocabulary of 105 knowledge keywords, with the earth ecosystem topic at high school-level as the learning domain [5]. The system has been placed on “the Web.” The learning screen (see Fig. 1) is comprised of three areas: for questions, for keyword searching, and for messages. The question area is used for offering questions based on text and/or images. Learners answer the questions by selecting answers from the selection items. The keyword searching area is used for outputting keyword descriptions and the necessary knowledge keyword list to the learner in accordance with the situation, so that the learner can acquire knowledge by answering the question. Finally, the message area is used for outputting the learning grade and results, as well as providing hints from the system.

2.1. Learning assistance mechanism

The concept of the learning assistance mechanism comes from the point that the relationship between the knowledge state of the learner and the answer correctness is established from the learning results, not based on rules, and is incorporated into the learning assistance mechanism on an individual basis. The block diagram of this mechanism is illustrated in Fig. 2. The total system is composed of three sections corresponding to the following functions: a question-setting assistance function based on evaluating the difficulty for each learner individually, an answering assistance function based on correctness prediction using case-based reasoning, and a learning system information-update function.

![Fig. 1. Question-driven learning system.](image-url)
can lead to a wrong answer. To realize the answer assistance function, answer prediction is performed by case base searching for the analogous case based on the learner’s knowledge state relevant to the current question. Based on the prediction result, the system offers assistance, e.g., by suggesting a keyword search. The learner is also provided with advice messages and a keyword list arranged in order of inadequacy of understanding as the guidance to keyword learning.

The learning system information is updated with reference to the learner’s knowledge model, the case database, and the keyword importance table for each question. The system updates the learner’s knowledge model after the computation of understanding levels for the relevant knowledge keywords based on his or her answer estimation, the acquired keywords and items, and the self-graded understanding levels. At this time, additional records for the newly learned keywords are also inserted with their understanding levels. The case database is updated based on the new learner’s knowledge model and the prediction results of the case-base search. The update process includes autoacquisition (recording), modification, and discarding (cancellation) of cases. For the searched cases, the prediction success ratio (case reliability) and prediction flags are modified, and those with low prediction-success ratios are discarded from the case base. Furthermore, with the updated case database, the system computes and updates the importance table for the keywords relevant to each question, based on the correlation between the understanding levels of each knowledge keyword and the answer correctness.

### 2.2. Learner’s knowledge model

The learner’s knowledge model describes what knowledge is understood by the learner, and to what extent, during his or her learning progress. The learner’s knowledge model $KB_s$ has the following data structure.

$$KB_s = (\text{Student#}, (K_1, R_1, \ldots, K_n, R_n))$$

In this formula, $\text{Student#}$ denotes the learner ID, $K_i$ denotes the learned knowledge keywords, and $R_i$ denotes the understanding levels with respect to these keywords. This knowledge model is constructed based on the learned content, the acquired keywords and items, self-graded understanding levels, and the answer correctness. For non-learned knowledge keywords, the system is unable to know in advance whether they are understood by the learner. However, an idea of information mastery based on group average understanding levels, or on the learner’s answers to the questions and his search histories for the knowledge keywords, can be useful. The learner’s knowledge model is
updated by inserting new records of newly acquired keywords and by updating the understanding levels for the existing keywords.

2.3. Case data structure

The set composed of the relevant knowledge keywords and the understanding levels required for answering the question is treated as the basic component of a case. It is assigned to each question in combination with the correctness flag and the prediction success ratio. The relevant knowledge keywords for each question are defined when the question is created by the teacher. The case data \( KB_q \) are generated based on the keyword list of the question’s relevant knowledge \( KB_q \).

\[
KB_q = (Question\#, K_1, K_2, K_3, \ldots, K_m) \quad (2)
\]

\[
KB_c = (Question\#, HitRatio, AnsFlag, (K_1, R_1) \ldots (K_n, R_n)) \quad (3)
\]

\( Question\# \) in these equations denotes question ID, \( K_i \) denotes relevant knowledge keywords, and \( R_i \) denotes understanding levels of the corresponding keywords. In addition, it is considered that within the understanding patterns of the keyword, there are ones that lead easily to the right answer and others that lead easily to a wrong answer. They are subdivided into the right-answer group and the wrong-answer group by defining a correctness flag, \( AnsFlag \). Moreover, the prediction success ratio, which is the ratio of the number of success predictions to the total number of predictions, is defined by \( HitRatio \).

2.4. Importance table of relevant knowledge keywords

Generally, it is unlikely that all the knowledge keywords relevant to the question are equally important to the right answer. Therefore, more reasonable results for searching analogous cases can be expected by assigning weights to each keyword in terms of its contribution to the right answer. The importance table \( KB_p \) for the relevant knowledge keywords is defined by the keyword \( K_i \) and importance weight \( W_i \) for each question \( Question\# \).

\[
KB_p = (Question\#, (K_1, W_1) \ldots (K_m, W_m)) \quad (4)
\]

Relevant knowledge keywords and the initial importance table are pre-set when the question is created by the teacher. After the case base is stored, the importance of each keyword is auto-computed based on the correlation between the keyword’s understanding levels and the result correctness. The computation method will be explained in the Appendix.

3. Question-Setting Assistance Based on the Generation of Individual Question Series

The question-setting assistance function, based on individually evaluating question difficulties, includes the operation of extracting the learner’s knowledge states regarding the questions, the operation of case-base searching that takes into account relevant keyword importance and case reliability, and the operation of generating the question series corresponding to the knowledge state of the learner.

3.1. Extracting the knowledge state for the question

The learner’s knowledge state regarding the question is a subset required for answering each question within his or her knowledge model. Usually, all the necessary information for the knowledge state regarding the question can be found in the learner’s knowledge model, but when non-learned knowledge keywords exist, their default understanding levels \( R_i \) are defined by the following formula, using the keywords understanding levels in those cases that belong to the current question. Within Eq. (5), \( y_j \) denotes the prediction success number of the case \( j \), \( R_{ij} \) denotes the understanding level of the keyword \( i \) in the case \( j \), and \( n \) denotes the total number of case data that belong to the current question. This value is equivalent to the group average of the learners who learned the current question up to that time:

\[
\bar{R}_i = \frac{\sum_{j=1}^{n} y_j \times R_{ij}}{\sum_{j=1}^{n} y_j} \quad (5)
\]

Using the information on the relevant knowledge keywords for each question, the system searches the learner’s knowledge model and extracts the understanding levels for these keywords. The knowledge state regarding the question is generated using all the relevant knowledge keywords, the keyword understanding levels, and the keyword importance table. Assuming that the learner’s knowledge model is \( KB_p \), the importance table for the relevant knowledge keywords is \( KB_p \), and the knowledge state regarding the question is \( KB_p \), the process of extracting the knowledge state is described below. Since non-learned keywords exist (see Fig. 3), the understanding level \( R_{10} \) regarding the current keyword \( K_{10} \) is computed by using Eq. (5).
value is $D = Sim$, and in the case of $V = 1.0$, the estimated value is $D = Sim/2.0$.

With the results of the case-base search, the system extracts one by one the cases that are most analogous to the learner's knowledge state for the corresponding non-learned questions, and generates two groups (right answer group AT and wrong answer group AF) comprised of the set of questions $Q_i$ and estimated values $D_i$. In the case when the prediction flag AnsFlag for the extracted case points to the right answer, this question is placed into the right answer group, and if the AnsFlag points to a wrong answer, it is placed into the wrong answer group.

3.3. Question series generation

Questions belonging to the right answer group $A_T$ and the wrong answer group $A_F$, respectively, are sorted in order of smaller estimated values $D_i$, and the two question series $A_T$ and $A_F$ are generated with difficulties corresponding to the learner's knowledge state at the moment. For the right answer group, the smaller the estimated value $D_i$, the higher the right-answer probability (lower difficulty), and the larger $D_i$, the lower the right-answer probability (higher difficulty). For the wrong answer group, the smaller the estimated value $D_i$, the higher the wrong-answer probability (higher difficulty), and the larger $D_i$, the lower the wrong-answer probability (lower difficulty).

Right-answer series $A'_T$

(small estimated values) $\Leftrightarrow$ (large estimated values)

easy $\Leftrightarrow$ difficult

Wrong-answer series $A'_F$

(small estimated values) $\Leftrightarrow$ (large estimated values)

difficult $\Leftrightarrow$ easy

This information is used to realize a teaching strategy that would allow selection of appropriate questions corresponding to the learner's knowledge state. Figure 4 illustrates the operation flow chart for the question-setting assistance function based on the generation of individual question series.

4. Question-Answering Assistance Based on Correctness Prediction

Upon being offered a question by the learning system, the learner begins vocabulary learning of the relevant knowledge keywords to answer the question. Before the learner has finished the keyword learning and started with the question-answering process, the learning system performs a correctness prediction for his answer by searching the case base using the current learner's knowledge state.
The question-answering assistance is accomplished by offering appropriate assistance messages and learning hints according to the prediction results.

To realize this process, the system extracts the learner's knowledge state as described in section 3.1, and performs the case-base search as described in section 3.2. In the first step, the case data stored with the current question is searched, and the case that possesses the smallest estimated value $D$ is extracted as the target case. However, this target case probably does not mean the optimum solution since its results from a search within the case data of the current question. There are possibly more suitable cases that exist within the case data of the other questions. Therefore, the search result with estimated value $D$ smaller than a certain threshold value is treated as the optimum solution, meaning that the search succeeded. Otherwise, the search is considered to have failed, and is repeated for another question's case data.

To repeat the search, the system must first identify which question among the other questions is the most analogous to the current one, and then search its case data. For searching through the analogous questions, the system uses the similarity, based on the relevant knowledge keywords and the importance table, between the current question and the others. Here, the similarity estimate is defined by the cosine of the importance vector angle in the question's relevant keywords space.

$$U_i = \frac{W_0 \cdot W_i}{||W_0|| \cdot ||W_i||}, \quad (i = 1, 2, \ldots, n) \quad (8)$$

In this formula, $W_0$ denotes the keyword importance vector for the current question, and $W_i$ denotes the keyword importance vector for the other questions. By calculating all $U_i$ using this formula, treating the question with the largest estimated value as the analogous question, and searching this question's case data, the correctness for the current question can be predicted.

Figure 5 illustrates the basic operations for this function. First, the case data of the current question are searched, and if the search fails or if no case data are present, an analogous question is to be identified. Afterwards, the case data of the analogous question are searched. If both searches fail, a wrong answer is predicted. The learner is offered the knowledge keywords in order of importance, and is advised to start vocabulary-searching before answering the question. If any of the searches succeed, the right answer or wrong answer is predicted according to the right/wrong prediction flag of the acquired case. If the right answer has been predicted, the learner is guided to the answering process by a message prompting him to answer carefully. If the wrong answer has been predicted, the relevant knowledge keywords are offered in the order of inadequacy of understanding to the learner. Also, the learner is guided to the keyword learning process by a message advising him to review the necessary knowledge before answering the question. After the learner has finished answering the question, the prediction success ratio and flag of the acquired case are modified according to whether the prediction had been successful. At this time, modification of the acquired case and correction of the case data items differ depending on whether the search was primary or secondary, and whether it succeeded or failed. In the case of having done the keyword learning, the case-base search must be repeated after updating the learner's knowledge model, and the newly acquired target case is

![Fig. 5. Correctness prediction-based answering assistance.](image-url)
modified. Details on the case acquisition, modification, and cancellation will be discussed in section 6.

5. Learner’s Knowledge Model Update

After the learner has finished the keyword learning or the system has estimated the learner’s answer, the learner’s knowledge model is updated. This operation involves adding knowledge keywords and modifying the understanding levels. The update procedure differs depending on whether the answer was correct, and on whether the learner performed the keyword learning. The table in Fig. 6 shows the update method.

If keyword learning was performed, the understanding level $R'_i$ is recalculated after the grade values for each item of the relevant knowledge keywords have been modified depending on whether the answer to the current question was right or not. Figure 6 shows the method for modifying the learner’s grade values. In the case of the right answer, the grade value is increased by one step, and for the wrong answer, the grade value is reduced by one step. The new understanding level is computed according to Eq. (9).

Considering that the information volume included in each item of keyword $i$ may be different, computation is conducted after assigning weights $a_{ij}$ to the grade value $M_{ij}$ for each item. These weights $a_{ij}$ are set by the person responsible for preparing the teaching materials:

$$R'_i = \sum_{j=1}^{n} a_{ij} \times M_{ij} \quad (9)$$

When the keyword learning was performed before answering the question, if the answer is right, the understanding levels for the relevant keywords are increased, and if it is wrong, these levels are reduced. The increase/reduction amount $\Delta$ is calculated according to Eq. (10) based on the importance value of each knowledge keyword. The new understanding level $R'_i$ of a keyword $i$ is a summation of previous understanding level $R_i$ and the increment $\Delta$. The value of $\Delta$ is a positive number for the right answer and a negative number for a wrong answer. The variation range of $\Delta$ is controlled by the importance $W_i$ of the keyword $i$, i.e., by its contribution to the question answering. In addition, the final obtained understanding levels are normalized to fit into the range (0.0 to 1.0).

$$\text{Understanding level: } R'_i = \begin{cases} R_i + \Delta, & (0.0 < R_i + \Delta < 1.0) \\ 1.0, & (R_i + \Delta \geq 1.0) \\ 0.0, & (R_i + \Delta \leq 0.0) \end{cases} \quad (10)$$

for a right answer; $\Delta = (1.0 - R_i) \times W_i$
for a wrong answer; $\Delta = (0.0 - R_i) \times W_i$

6. Case Database Update

Upon updating the learner’s knowledge model after the question has been answered, the system performs manipulation of the case base, including case acquisition, modification, and cancellation. The newly obtained case data are inserted into the case base of the corresponding question, and the modified (corrected) case data are substituted for the previous ones. The canceled (removed) case data are discarded from the case base.

If the case-base search for the correctness prediction has failed, the knowledge state corresponding to the current question is extracted from the learner’s knowledge model after the learner has answered the question, and acquired as a new case datum. The keyword understanding levels $R_i$ of these case data are computed using the equation in section 5, and the $\text{AnsFlags}$ are set depending on the answer correctness. The prediction success ratio $\text{HitRatio}$ is defined as Eq. (11), with the total number of predictions denoted by $Y$, and the number of successful predictions by $y$. For a new case, $Y$ equals 1, with $y$ set to 0 for the right answer and to 1 for the wrong answer.

$$\text{HitRatio} = \frac{\text{number of successful predictions}}{\text{total number of predictions}} = \frac{y}{Y} \quad (11)$$

The operation for the cases is different depending on the primary search success, secondary search success, and the search failure in case searching. The process contents, as well as the update data, the reference data, and the items
are summarized as a table in Fig. 7. $KB^r_i$ and $KB^f_i$ denote, respectively, the updated learner’s knowledge model and the knowledge state for the current question. $KB^r_i$ denotes the case data newly found based on the new knowledge state, and $KB^f_i$ denotes the newly acquired case data.

If a right answer is predicted in the primary case-base search, the system updates the total number of predictions and the number of successful predictions depending on the learner’s actual correctness, and computes HitRatio. As for the understanding levels, the new understanding levels $R'^i$ are computed based on the understanding levels $R^i$ obtained from the learner’s knowledge state and those $R^c_i$ in the acquired case with considering the total prediction number ($Y$):

$$R'^i = \frac{R^i + Y \times R^c_i}{Y + 1} \quad (12)$$

If a right answer is predicted in the secondary case-base search, the acquired case must be copied into the current question’s case base, since this datum belongs to a different question, and then HitRatio, $R^c_i$, and AnsFlag are calculated with the method mentioned above. Since a different question’s case base is being used, it is assumed that some keywords might have been omitted. Therefore, the understanding levels of the omitted keywords should be obtained from the learner’s knowledge model and incorporated into the copied case.

If a wrong answer is predicted from the case-base search results, considering that the learner starts answering the question after having performed the keyword learning, the learner’s knowledge model must be changed as compared to that before the case-base search. Therefore, case-base search must be re-executed after obtaining a new knowledge state from the changed learner knowledge model. The update is applied to the newly found case data, rather than to the one used for prediction. In the case of a learner’s answer result matching the case-data prediction flag, only the prediction success ratio HitRatio and the understanding levels $R_i$ need to be updated. If there is no match, in addition to the above-mentioned items, the prediction flag AnsFlag is also updated based on the updated value of the prediction success ratio. Since the prediction success ratio HitRatio is a ratio of the number of successful predictions to the total number of predictions, and moreover, since the prediction flag AnsFlag defines only whether the answer is right or wrong, the case-prediction success ratio lower than 50% means that the case must to be used for opposite prediction. In other words, if the success ratio for the case with the right-answer prediction flag does not exceed 50%, the success prediction probability will be higher when the wrong answer prediction is made, rather than the right answer. In this case, the prediction flag AnsFlag is changed from the right answer to the wrong answer, or from the wrong answer to the right answer, and then the prediction success number is recalculated. Thus, the equation $0.5 \leq \text{HitRatio} \leq 1.0$ is usually true.

$$y_f = Y - y_i \quad (13)$$

$y_f$: number of successful predictions of wrong answer
$y_i$: number of successful predictions of right answer

Moreover, among the cases with the prediction number ($Y$) exceeding a certain number, the cases with the success ratio close to 50% are removed (discarded) from the case database. The fact of the success ratio being 50% means the 50–50 possibility that the learner with such knowledge state matching the case data would provide the right or wrong answer to the question. These case data are considered not effective for predicting the answer to this question. The threshold value for canceling the cases with the prediction success ratio close to 50% is not defined by any literature. No standard would guarantee the reliability if the success ratio is greater than some value. However, the number of remaining cases beyond the total case data can be set to some figure and be used as the cancellation condition for deciding the prediction success ratio threshold. In this regard, it should be noted that:

- To preserve a meaningful prediction success ratio, the total prediction number of the corresponding case data must exceed a certain number. Therefore, the case data with a prediction number smaller than the certain number must be kept, regardless of the prediction success ratio.
- For the case data with a sufficient prediction number, there is a preset number of cases remained beyond the time limit required for the case-base

![Fig. 7. List of operations for updating the case base.](image-url)
search. The rest of the cases are set for preferential
cancellation in order of the success ratio close to
50%.

In our learning system, the minimum prediction num-
ber was set to 100 and the maximum number of cases was
set to 256.

7. Estimation Test

The proposed learning assistance mechanism was
incorporated into the exercise-based learning system
(SEBLE: Supportive Exercise-based Learning Environ-
ment) created by the authors, and the performance of each
function was evaluated. The system was implemented on a
workstation SS-2 (SUN SPARC-2), where the learning
client section was coded using JAVA for deliverability on
the widely used Netscape platform. The learning informa-
tion server was implemented using JAVA and C-language.
Each function of this learning assistance mechanism was
implemented using C-language.

Nine learning themes related to “The Basis of Eco-
systems and Their Environmental Influence” within the
world environment field were used as the learning content,
and measurements were made regarding one of them, “Eco-
system Structure.” The topic “The Basis of Ecosystems and
Their Environmental Influence” included 105 knowledge
keywords, and the learner was offered definitions for each
keyword in the form of vocabulary, to which he or she was
able to refer without restriction. The theme “Ecosystem
Structure” included 12 questions associated with relevant
knowledge keywords from 5 to 13.

The test involved 10 learners who learned the same
theme—“Ecosystem Structure.” We measured the case data
number accumulated for each question, and time spent for
computations and other operations required for the question
arrangement in sequence. The measurement results are
summarized in Table 1.

Time required for question sequencing is a function
of the total number of the case data relevant to all questions
in the theme, and time required for importance value cal-
culation and correctness prediction is a function of the number
of the case data relevant to each question, respectively.
Therefore, time consumption in all processing procedures
can be reduced by decreasing the number of cases to search
in a way of allocating the case data to each question.
Moreover, our algorithm ensures that the relevant keywords
importance table is recalculated only when the current
question’s case data are being modified, and that even
during the correctness prediction only the current ques-
tion’s case data are searched. This is so the entire system’s
response can be accelerated. As shown in Table 1, we have
confirmed that the response time that the learner can be
offered the next question is within 2 s, correctness can be
predicted in little more than 1 s after setting the question,
and the importance table can be recalculated within 1 s.

All the processes in the proposed learning assistance
mechanism need to involve at least several case data for
each question. If there are too few cases, the accuracy of the
correction prediction and importance calculation will be
poor, and the reliability of the question series difficulty will
be low. With the increasing number of the case data due to
the use by many learners, although the accuracy of each
process becomes higher, the total system response becomes
sluggish due to more cases to be searched. In our system,
in addition to the cancellation operation of the non-effective
cases, it is possible to control the number of case data by
varying the similarity used as a threshold for the new case
acquisition.

8. Conclusions

In this paper, the authors proposed a learning assis-
tance mechanism using case-base reasoning. Considering
that even the same question stands for different difficulty to
different learners, we introduced the question-setting assis-
tance function, which involves dynamically evaluating the

<table>
<thead>
<tr>
<th>Measurement item</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>KW number in learner’s knowledge model</td>
<td>34</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>Relevant knowledge KW number for each question</td>
<td>13</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Case data number for each question</td>
<td>213</td>
<td>44</td>
<td>164</td>
</tr>
<tr>
<td>Time required for question sequencing (ms)</td>
<td>1835</td>
<td>1673</td>
<td>1713</td>
</tr>
<tr>
<td>Time required for correctness prediction (ms)</td>
<td>1169</td>
<td>237</td>
<td>706</td>
</tr>
<tr>
<td>Time required for importance degree calculation (ms)</td>
<td>987</td>
<td>190</td>
<td>470</td>
</tr>
</tbody>
</table>
non-learned questions difficulty based on the similarity by the understanding levels for the question's relevant knowledge and then arranging the questions in sequence. We also discussed the answering assistance function, which involves predicting the answer correctness based on the student's knowledge state and using the prediction results to offer an easy keyword learning environment.

The functions provided for this system, such as the case-base auto-update function and the learner's knowledge model auto-update function, contribute to improved reasoning reliability due to the refined case base from analogous case data. In addition, the function of deriving the relevant keywords' importance table using the case data accounts for improved reliability due to the number of case data increasing with more participating learners. Thus, the importance table that is subjectively defined by the teacher can be objectively verified in the process.

While building and using the case base, one must take into account the response time issue. With the framework of our system, we were able to achieve control over the entire case collection due to intensive use of analogous cases and to accelerate response due to reducing the number of cases to be searched by allocating cases to each question. According to the test results, response time for both the question series generation and the correctness prediction was about 1 to 2 s.

Our future work will include developing methods for evaluating the results of using the reasoning methods and studying this framework's efficiency based on educational technology theory. In addition, we are planning an advanced study aimed at objective estimations on the understanding levels derived from the correctness of the learner's answer, and on those derived based on the learner's self-grading that was used in each function above.

REFERENCES


APPENDIX

Calculation of importance table for question-relevant keywords

To calculate the importance values of the relevant keywords for a question, the ID3 method [6, 7] is used. Upon extracting the sample data $E_p$ from the $n$ case data, the corresponding understanding levels $R_i$ and correctness flag $AnsFlag$ are allocated to $m$ keywords for each $E_p$, respectively.

$$E_p = ((K_1, R_1), \ldots, (K_m, R_m), AnsFlag)$$

In a simplified ID3 method, we use the following calculation procedure based on a viewpoint in accordance with the relevant keyword contribution to the answer correctness.

- All sample data are distributed into the right-answer and wrong-answer groups based on the correctness flag $AnsFlag$.
- Each keyword's understanding level is assigned a binary value based on the mean understanding level: 1 corresponds to "understood," 0 corresponds to "not understood."
- The information volume $S$ is introduced in accordance with the ID3 method. Total information volume (total entropy) is calculated by the following formula. $p$ denotes the ratio of data that belongs to the right-answer group among the all sample data, $q$ is the ratio of data that belongs to the wrong-answer group.

$$S = -p \times \log_2 p - q \times \log_2 q$$

- Information volume (entropy) is calculated for each $R_i$ of the keyword. $p_i$ denotes the ratio of data which belongs to the right-answer group among the sample data with the binary understanding level $\alpha = (0, 1)$ for the keyword $K_i$. $q_i$ denotes the ratio of data which belongs to the wrong-answer group.
\[ S_i^a = -p_i^a \times \log_2 p_i^a - q_i^a \times \log_2 q_i^a \]

- Mean value of the information volume (entropy) is calculated for each \( R_i \) of the keyword. \( P_i^a \) denotes the ratio of the sample data with the binary understanding level \( \alpha = (0, 1) \) for the keyword \( K_i \).

\[
S_i = \sum_{\alpha=0}^1 S_i^a \times P_i^a
\]

- Difference between the mean information volume (entropy) and the total information volume (total entropy) is calculated for each \( R_i \) of the keyword.

\[ \Delta S_i = S - S_i^+ \]

The obtained value is the information volume regarding the answer correctness that reflects whether the keyword was or was not understood. It means that the larger the value \( \Delta S_i \) the greater the information volume obtained from the keyword \( K_i \). In other words, it is shown that the value \( \Delta S_i \) is a measure of importance for answering the current question correctly. We define this value as the relevant keyword importance.

**Authors** (from left to right)

Ban Kaku (member) received his B.E. degree and M.E. degree in Computer Science from Shenyang Institute of Technology, China, in 1982 and 1984, respectively. In 1988, he received his Ph.D. in Information Engineering from Hokkaido University, Japan. He joined the Software Research Laboratory in 1991, and his research interests include artificial intelligence, digital image processing, and educational technology. He is a member of the Institute of Electronics, Information and Communications Engineers of Japan.

Tetsuya Matsumoto (member) graduated from Hokkaido Normal School in 1948, and from the Electronics Dept. of the Engineering Faculty, Hokkaido University, Japan, in 1964. He became Supervisor of the Hokkaido Education Committee in 1967, and the part-time teacher at the Hokkaido Education University in 1976. He joined the Software Research Laboratory in 1991, and is studying at Intelligent CAI. He is a member of the Institute of Electronics, Information and Communications Engineers of Japan.

Norimichi Kojo graduated from the Applied Physics Dept. of the Engineering Faculty, Hokkaido University, Japan, in 1969. He received his Ph.D. in physics from Tokyo State University in 1977. He joined the Software Research Laboratory in 1991, and is studying at Intelligent CAI. He is a member of the Information Processing Society, and Physics Society.

Guo Xin (photo not shown) graduated from Shenyang Teacher Academy of China in 1991 and from the graduate school of North-East University, China, in 1992, where she received her Master’s in Education. She joined Shenyang Administration College of China as a professor in 1991, and her research interests include artificial intelligence, computer networks, and educational technology. She is an Executive Director of the Institute of Computer Education Technology in Higher Education of China.