A Learning Assistance Tool for Enhancing ICT Literacy of Elementary School Students

Chenn-Jung Huang, Ming-Chou Liu, Kuo-En Chang, Yao-Ting Sung, Tz-Hau Huang, Chun-Hua Chen, Hung-Yen Shen, Kuo-Liang Huang, Jia-Jian Liao, Kai-Wen Hu, Yun-Cheng Luo and Tun-Yu Chang

1Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan // cjhuang@mail.ndhu.edu.tw
2Department of Curriculum Design and Human Potentials Development, National Dong Hwa University, Taiwan
3National Taiwan Normal University, Taiwan
4Graduate Institute of Networking and Multimedia Technology, National Dong Hwa University, Taiwan
5Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
6Department of Computer Science, National Tsing Hua University, Taiwan
7Department of Resources Engineering, National Cheng Kung University, Taiwan

ABSTRACT

With rapid advances in the development of information technology, information access has become central to life in the 21st century. In recent years, the development of useful learning-assistance systems has become a popular topic in literature. Learners can benefit from guidance provided by a tool that assists learning when a student has trouble using an e-learning platform. An effective learning-assistance tool can reduce teacher workload and increase the time spent giving individual guidance to learners who fall behind. However, few machine-learning techniques have been used as learning-assistance tools in literature to determine learner status or provide feedback to learners and teachers. Therefore, this work employs a learning-assistance tool that uses learning-reinforcement techniques to continuously interact with an environment and provides learners with suitable and timely feedback to guide students through difficulties, inspire them, and help them complete assigned tasks. The learning-assistance tool can promote learners’ information and communication technology (ICT) literacy and assist learners in overcoming difficulties. Furthermore, teacher workload is significantly reduced because appropriate hints or feedback are automatically delivered to learners without teacher involvement.

Keywords

Information and communication technology, Learning assistant, Diagnosis, Machine learning, Reinforcement learning

Introduction

Internet learning via e-learning, distance education, remote experimentation, and computerized examination has been extensively used at all education levels. Numerous web-based e-learning platforms and examination systems have been developed. These information and communication technology (ICT)-based systems reduce the time needed for examination-related tasks, such as grading, marking, recording, and analyzing, and provide ubiquitous practice and tests via web browsers.

As students can learn effectively with and be encouraged by a learning-assistance tool that provides useful hints or feedback, this work developed a novel learning-assistance tool that guides learners when they become confused or stalled when using an ICT literacy-assessment platform constructed for elementary school students in Taiwan. Project-based activities are incorporated into the web-based diagnostic and assessment platform. Students are expected to use the application software suite provided by the platform to solve everyday problems. Guidance is offered via a feedback rule construction mechanism. A machine-learning technique, reinforcement learning, is adopted to provide useful hints or feedback to learners based on their learning portfolios, collected during learning activities. To the best of our knowledge, this is the first diagnostic and assessment tool in literature. Use of Asynchronous Java Script and XML (AJAX) technology in this work enhances interactivity, access speed, and usability of the web-based diagnostic and assessment platform, such that problems associated with traditional click-and-wait web applications can be alleviated. That is, via AJAX technology, the web-based diagnostic and assessment platform can respond promptly to ensure timely interaction between the platform and learners. According to
experimental results, the proposed learning-assistance tool effectively assists learners in making progress during continuous assigned projects. The assessment module was confirmed capable of evaluating learner work quality correctly.

The remainder of the paper is organized as follows. Related work in literature is discussed in the next section. Following that, the overall architecture of the proposed ICT learning-assistance tool, how to apply the reinforcement learning mechanism to the learning diagnostic module, and experimental results are discussed in their respective sections. Finally, conclusions and future work directions are given.

Related work

In the late 1990s, many governments increased their investments in ICT in the educational domain. The rapid development of the Internet and the World Wide Web led to the adoption of plans to equip all schools with accessibility to these facilities within a short period. In England, the government launched a six-year project to build an information-technology platform for delivering onscreen tests to all secondary schools. The tests were developed to enable students to apply technologies to solve problems, analyze information, develop ideas, create models, and exchange information (Becta, 2006). A computer-proficiency test developed by Bradlow, Hoch, and Hutchinson (2002) consisted of 27 multiple-choice items measuring student computer knowledge in several domains, such as general computer terminology, file management, spreadsheets, databases, the Internet, and email. However, the test is not specifically designed to test the digital skills needed by primary school students. The items measured factual knowledge of computer terminology and concepts rather than gauging procedural knowledge about how to perform computer tasks. In Taiwan, several ICT-assessment platforms have been constructed to assess the computer knowledge of primary school students via multiple-choice questions. Students typically memorized answers to those multiple-choice questions, such that the multiple-choice questions did not assess student ICT literacy.

A computer-based performance test in which students are assessed via a number of functional computer tasks is undoubtedly more effective for assessing digital skills than a questionnaire. All levels of operation, such as response times, mouse-click sequences, and other user actions, could be automatically recorded in computer-based performance tests. Recorded behaviors facilitate in-depth analysis of student digital skills. Kuhlemeier (2007) demonstrated that computer-based assessment has tremendous potential to learn about what students know across various ICT environments.

After investigating the concepts of ICT-literacy assessment in England and the USA, project-based activities are incorporated into a novel assessment platform to meet the pressing need of Taiwan’s nine-year mandatory school program. Students are expected to apply an application software suite provided by the platform to solve daily life problems. Sequentially, the assessment platform grades student outcomes and generates statistics for both teachers and students.

In the last few years, development of useful learning-assistance systems has become a common research topic in literature. ActiveMath, a web-based mathematics learning-assistance system (Melis, Gogua, Homik, Libbrecht, Ullrich, & Winterstein, 2006), assists learners in searching for interesting courses or practical examples of mathematics. The system can intelligently analyze learner input and provide appropriate and timely feedback to learners when they make mistakes in completing math problems. The learning-assistance system was helpful in stopping learners from wasting time on unnecessary mistakes, and effectively improved learner achievement. Huang, Chen, Luo, Chen, and Chuang, (2008) incorporated a diagnostic and assessment tool into an e-learning platform developed for programming language courses. The proposed learning diagnosis assessment tools based on text-mining and machine-learning techniques were employed to reduce teacher workload. Moreover, Huang, Chen, and Chen, (2009) proposed an argumentation processing agent for computer-supported cooperative learning. Learners are first assigned to heterogeneous groups based on a questionnaire given directly before they start learning activities on the e-learning platform. The argumentation-processing agent then analyzes the learning portfolio of each learner in an e-learning platform and automatically issues feedback in cases of a poor argument or abnormal behavior.

All of the above-mentioned learning-assistance systems employed machine-learning techniques to provide assistance to learners and teachers. Thus, this work adopted a well-known machine-learning technique, reinforcement learning, to track learner operations and provide suggestions to learners when learners were confused or stalled when using a
previously developed ICT-literacy assessment platform. The AJAX technique is employed to ensure timely interaction between a user and the platform. One key advantage of using AJAX is that web standards in AJAX are well defined and supported by all major browsers.

In recent years, many developers have built web applications using AJAX technology. Expedia added features such as pop-up calendars on its travel site using AJAX (Paulson, 2005). Google worked with AJAX to construct applications such as Gmail and Google Groups, a community and discussion service. Flickr utilized AJAX on some of its websites, allowing users to add and view photos (Patrick, 2006). Additionally, AJAX-based Google Maps allows users to hold down the left mouse button and slide the cursor over an image to retrieve part of a map not shown on the screen. Updates occur smoothly and images move and change immediately. With typical web applications, users must wait for entire pages to reload, even when image changes are small (Wang, & Bian, 2007). A synchronous learning environment, called the Synchronous Learning Environment with Web 2.0 (SLEW) (Lin, Chi, Chang, Cheng, & Huang, 2007), has a course agent, clear user interface, and an interaction mechanism for teachers and learners developed using AJAX technology. SLEW can support teachers and students who are participating in synchronous learning via poor network bandwidth. Furthermore, AJAX technology is employed in SLEW to retain learning materials, such that learners can review their materials during learning processes. The most important feature of SLEW is that AJAX technology is applied to partially update a web page rather than re-loading all page content.

Reinforcement learning is to study how animals and artificial systems learn to optimize their behavior when provided with rewards and punishments. Reinforcement learning algorithms have been developed that are closely related to dynamic programming methods, which are general approaches to optimize control. Reinforcement learning is a novel approach to systems management that differs radically from conventional model-building approaches. Van Vliet, Kletke, and Chakraborty, (1994) developed a reinforcement learning agent that can learn to play the Othello game without using knowledge from human experts. Their experiments demonstrated that a player employing reinforcement learning agents to learn how to play Othello outperformed players using basic strategies. To the best of our knowledge, a reinforcement learning technique has never been applied to the design of an intelligent e-learning platform.

![Figure 1. Architecture of learning assistance tool](image-url)
Architecture of learning-assistance tool

Figure 1 shows the architecture of the learning-assistance tool in this work. The e-learning platform has four main components — a script content/learning activity-management module, learner workspace interface, a learning diagnostic module, and an assessment module. The script content/learning activity-management module is used by teachers to construct and modify script content for students using the learning-assistance platform. The student workspace interface has a system control area that provides such functionalities as login/logout and calling for help. Step-by-step instructions and hint messages can be displayed for students. Five application software module selection buttons and a software workspace are also in the student workspace interface. The five application software modules developed in this work possess functions similar and compatible to those in file browsers, Word, Excel, and PowerPoint. Three open-source software modules are modified to mimic the functions provided by the Microsoft application suite — Word, Excel, and PowerPoint. Furthermore, two additional tools, a built-in search engine and an email manager, are also developed.

The inputs for the learning diagnostic module are obtained from learner portfolios updated during online learning activities on the e-learning platform. Learner portfolios include an assessment of each learner, provided by the teacher based on learning records and response times on the e-learning platform. The diagnostic module monitors learning status and provides timely assistance to students who require help. The assessment module evaluates student learning outcomes using learning portfolios and transfers final reports to teachers and students.

The assessment module

The automatic assessment module has two main components, an expected result generator and a comparison unit. The expected result generator assists the teacher who designed a project to generate an expected result after each step. The expected results are converted into tags and saved in the database. The comparison unit merely compares student answers with expected answers saved in the database.

For instance, a teacher may expect that students can use the network search engine to search for information of the habitual records of the animals. The teacher is asked to design related web pages and upload them to the system before class to limit the material students can search and to ensure the quality of automatic assessment. Students are required to collect and organize information associated with an animal’s special habits from the web using the word-processing module. Each learner then assigns a file name to her/his file, which is defined by the script content, and sends it to the teacher as an email attachment. The entire process can determine whether students can use the network search engine and word-processing software and send attachments by email. This script can assess whether students are capable of extracting information from websites and indicate whether the material source is abiding by copyright laws. The automatic assessment system designed in this work examines learner achievements for this example via the following steps:

- Determine whether students can use the search engine to search the website and access the specific website related to assigned task.
- Compare the file edited by learners using the word-processing module with the expected answer given by an expert to determine if learners pass this assessment.
- Examine whether students are capable of sending email attachments.

The learning diagnosis module

A feedback guidance mechanism is developed to continually give learners timely adaptive feedback messages to enhance learning achievement according to learning portfolios. The learning diagnostic module, which is embedded in the reinforcement learning mechanism, offers appropriate feedback to enhance learning outcomes. The learning diagnostic module provides a reference for teachers or the proposed system and gives the necessary assistance to students when they encounter a problem during a specific step. As the reward value is lower than a preset threshold, the learning-assistance mechanism gives appropriate feedback based on the environment and reward setting. Notably, the threshold is first determined by several experienced elementary school teachers who are familiar with student information literacy levels, based on the complexity of the experiment script and time limitation to finish the
test. The threshold can be adjusted during the prototype phase to reduce any gap between teacher expectations and student performance.

**Application of reinforcement learning mechanism to learning diagnosis module**

Reinforcement learning is learning how to take appropriate actions such that a numerical reward is maximized. Trial-and-error search and the reward value are important features of a reinforcement learning mechanism. An agent must develop what it knows to obtain a reward, and explore to make superior action choices in the future. With the reinforcement learning mechanism, a learning agent can improve its performance using feedback from the environment. Environmental feedback is called the reward signal. Reinforcement learning differs from supervised learning in how an output error is treated. Feedback with supervised learning shows information needed for an exact output, whereas feedback with reinforcement learning merely contains information about the quality of output.

Notably, Q-learning (Rahimiyan & Mashhadi, 2008) is a recent form of the reinforcement learning technique and is primarily concerned with estimating an evaluation of performing specific actions at each state. Thus, Q-learning has many successful applications, such as bidding in a power market, mapping and navigating the surface of Mars, clearing hazardous waste, and conducting rescue missions following earthquakes. Since Q-learning is a model-free algorithm that can be implemented easily, it is modified as the Q-learning algorithm in this work to enhance decision-making for the learning diagnostic module.

Figure 2 shows the normal operation of reinforcement learning. An agent learns effective decision-making policies via an online trial-and-error process in which the agent interacts with an environment. Each interaction consists of the following:

- Observing an environment’s current state $s_t \in S$ at time $t$, where $S$ is the set of possible states
- Performing some legal action at state $s_t$
- Receiving a reward $r_t$, which is a numerical value a user would like to maximize, followed by an observed transition to a new state, $s_{t+1}$

![Figure 1. Reinforcement learning mechanism](image)

Figure 2 shows the framework of reinforcement learning mechanism, which is flexible and can be applied in different ways. For example, time steps do not need to refer to fixed time intervals. They can, rather, refer to arbitrary successive stages of decision-making and acting. Actions can be low-level controls, such as voltage applied to the motors for a robotic arm, or high-level decisions, such as the decision of whether or not to have lunch. Therefore, actions can be any decision in which a learner attempts to learn how to make a decision, and states can be anything learners know that may be useful when making decisions.

**Traditional $Q$-learning algorithm**

The aim of the $Q$-learning agent is to learn an optimal policy, that is, a mapping from a state to an action that maximizes expected discounted future reward, represented as a $Q$-function, which can be calculated recursively.
Thus, Q-learning algorithms work by estimating the values of state-action pairs. The value \( Q(s,a) \) is defined as the expected discounted sum of future payoffs obtained by taking action \( a \) from state \( s \) and following the current optimal policy thereafter. Once these values are learned, the optimal action from any state yields the result with the highest \( Q \)-value, as shown in Figure 3. The values for the state-action pairs are learned using the following \( Q \)-learning rule:

1. \[
\hat{Q}(s_t, a_t) = Q(s_t, a_t) + \eta (r_t + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)),
\]

where \( Q(s_t, a_t) \) is a value function defined for a state-action pair \((s_t, a_t)\) at time \( t \); \( \eta \) and \( \gamma \) are learning rate and discount factor, respectively; and \( r_t \) is a reward received as a result of taking action \( a_t \) in state \( s_t \).

For all states \( s_t \in S \) and all actions \( a_t \in A \) at time \( t \),
initialize \( \hat{Q}(s_t, a_t) \) to an arbitrary value
Repeat (for each trial)
  Initialize the current state \( s_t \)
  Repeat (for each step of trial)
    Observe the current state \( s_t \)
    Select an action \( a_t \) using a policy
    Execute action \( a_t \)
    Receive an immediate reward \( r_t \)
    Observe the resulting new state \( s_{t'} \)
  Update \( \hat{Q}(s_t, a_t) \) according to Eq. (1)
Until \( s_t \) is a terminal state

Figure 3. Q-learning algorithm

An extended Q-learning algorithm

An extension to the original Q-learning algorithm is proposed. First, an \( R \)-function is defined to represent the evaluation of state-action as follows:

\[
R = \sum_{k=1}^{n} \gamma^k \times r_k,
\]

where \( R \) is the general reinforcement learning value function. This function sets the probability of each path learners choose as \( \gamma \), and reward \( r \) is multiplied by \( \gamma \). Additionally, \( k \) is the step number of the path probability.

The probability of a user choosing an action from \( j \) actions in the \( z \)th state is

\[
p_{i,j} = \frac{n_j}{n_i},
\]

where \( n_i \) is the total number of users at the \( i \)th state, and \( n_j \) is the number of users choosing the \( j \)th action.

A threshold is set to determine whether a learner can finish an action within an expected time interval. If a learner fails to finish this mission within the expected time interval, a time reward is sent to the learner as follows:

\[
t_{\text{reward}} = \begin{cases} 
0, & \text{if } t < t_{\text{timeout}} \\
 t \times r, & \text{if } t > t_{\text{timeout}}
\end{cases},
\]

where \( t_{\text{timeout}} \) is a preset timeout, \( t \) is elapsed time after a timeout, and \( r \) is a time reward given by experts.

Total reward, defined as the selection of the best way to proceed, is expressed by
\[ R = \sum_{i=1}^{n} \left( \prod_{j=1}^{m} \max(p_{i,j}) \times \text{reward}_{i,j} \right) - t_{\text{reward}} \]

where \( \max(p_{i,j}) \) represents the most likely action, which is indexed by \( j \), selected during the \( i \)th stage; \( \text{reward}_{i,j} \) is the reward given by experts for each action node on a path; \( t_{\text{reward}} \) is subtracted per unit time in this equation because the probability of completing the mission decreases over time. When the total reward, as given by Equation 5, is smaller than the preset threshold determined by experts, a hint or suggestion is issued by the learning diagnostic module to guide learners. Three levels of hints or suggestions, including weak, moderate and strong, are provided. For example, in the mission that requires a learner to use the email module to send mail, a weak hint or suggestion, as shown in Figure 4, is issued to guide the learner when the learner gets stuck at an action node on an unexpected path for the first time. A moderate hint or suggestion, as given in Figure 5, is provided when the learner cannot get back on the right track after receiving a small hint. Finally, a strong hint that gives a direct instruction, as shown in Figure 6, is used to prompt learners when they continually go astray from expected paths.

Figure 4. An example of weak hint

Figure 5. An example of moderate hint

Figure 6. An example of strong hint

Figure 7 shows a diagram of the expected solution path for the example of sending email; the reward value of each action node defined by experts is included. In this example, the mission goal is to use the email module to send mail and receive a reply. Thus, the black action nodes along the middle path in Figure 7 should be chosen to complete the task. If a learner walks along the correct path in Figure 7, the system will give the student a positive reward provided by an expert based on the importance of action nodes to complete the mission. For example,

1. If learners launch the email module during the first stage, a learner likely grasps the key information or concept needed to solve the problem because launching the email module is the key to choosing the correct path to complete the mission. The learner can then receive a high positive reward value of, say, 15, accordingly. However, if the learner chooses other application modules, such as any of the blue nodes in Figure 7, the learner has no chance of completing the assigned task. The learner then receives a negative reward value of, say, \(-3\).

2. When the learners choose the correct path during the first stage, they will be led to the second stage. In the example shown in Figure 8, learner selection of the path during the second stage includes receiving a new email, sending email, replying, deleting, backing up the mail folder, and using the trash folder. If a learner chooses the right path during this stage, which is to send new email, the learner will receive a reward valued of 10. Other choices will give learners a negative reward value of \(-1\).

3. The reward values for nodes at other stages are assigned in a similar manner.
Figure 2. Illustration of expected solving paths for an email sending example

Figure 3. Workspace interface of email module
If a learner successfully solves the problem without following the expected paths designated by a teacher, the learner’s recorded operating sequence for the assigned project will be reported to the teacher to obtain teacher approval. The teacher will then specify the reward values for each action node on the solution paths constructed by the learner.

**Experimental results and analyses**

To verify the effectiveness of the reinforcement learning mechanism in the proposed learning-assistance platform, an operation script and two situation mission scripts were developed for elementary school students. An experimental group and control group were constructed, each consisting of 29 Grade 5 students. A pretest was given to all participants in the two groups to determine whether these students have the capabilities for file management, email, using an Internet browser, presentation, and word processing. Both the experimental group and the control group were given three tests. After being given the story background, students freely decided how to interact with the system to solve the assigned problems. No learning-assistance mechanism was provided to the control group during the assessment. The students in the control group were expected to find useful information on the assessment platform to complete the mission. Conversely, the experimental group was aided by the learning-assistance tool such that they were prompted with timely and appropriate hints or feedback. Furthermore, a post-test was given to both groups using the assessment platform without the learning-assistance tool.

Table 1 through 6 show $t$-test results of the pretest for the high-, medium-, and low-achievement categories in the experimental and control groups, respectively. Table 1 lists the means and standard deviations for the 10 students in the high-achievement category in the experimental and control groups, followed by differences between means. The mean and standard deviation for the 10 students in the high-achievement category in the experimental group were 97 and 4.83, respectively, while those for the 10 students in the control group were 95.5 and 4.97, respectively. Table 2 is used to determine whether the null hypothesis — no significant difference between mean scores for the two groups — can be accepted. The interpretation of the independent sample $t$-test of pretest is a two-stage process. The first stage involves examining the homogeneity of the variance between the two groups. The independent sample $t$-test analysis examines whether equal variances in the two compared groups can be assumed by using Levene’s test for equality of variances at the second stage. SPSS is adopted to compute both the $F$-statistic and $p$-value (% Sig.). If Sig. is less than 0.05 ($p < 0.05$), the Levene’s test indicates that the variances between the two groups are not equal. If Sig. is larger than 0.05 ($p > 0.05$), the Levene’s test indicates that equal variances can be assumed. Notably, Table 2 shows $F = 0.151$ and $p = 0.702$, which indicates that $p > 0.05$ and equal variances can be assumed. Thus, the null hypothesis — no significant difference between mean scores for the high-achievement category in the two groups — was tested using the $t$-test. The two-tailed significance ($p$-value) was 0.503. Hence, the difference between mean scores for the high-achievement category in the two groups was not significant at $p > 0.05$; thus, the null hypothesis is accepted. Since mean scores for the high-achievement category in the experimental group and control group were not significantly different, we infer that students in the high-achievement category in both groups have similar computer skills.

Table 1. Group statistics of pretest for the students in high-achievement category

<table>
<thead>
<tr>
<th>Group</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>The experimental group</td>
<td>10</td>
<td>97</td>
<td>4.83</td>
<td>1.52</td>
</tr>
<tr>
<td>The control group</td>
<td>10</td>
<td>95.5</td>
<td>4.97</td>
<td>1.57</td>
</tr>
</tbody>
</table>

Table 2. Independent sample $t$-test of pretest for the students in high-achievement category

<table>
<thead>
<tr>
<th>$F$</th>
<th>Sig.</th>
<th>$t$</th>
<th>$df$</th>
<th>Sig. (2-tailed)</th>
<th>Mean Difference</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.151</td>
<td>0.702</td>
<td>0.684</td>
<td>18</td>
<td>0.503</td>
<td>1.5</td>
<td>-3.10</td>
</tr>
</tbody>
</table>

CI = Confidence Interval, $df$ = degrees of freedom

Table 3 lists the means and standard deviations for the 10 students in medium-achievement category in both groups, followed by difference between means. The mean and standard deviation for the 10 students in the experimental group were 81 and 5.16, respectively, while those for the 10 students in the control group were 79 and 5.16, respectively.
Table 4 shows \( F = 0.443 \) and Sig. \( p =0.514 \), indicating that \( p > 0.05 \) and equal variances can be assumed. Thus, the null hypothesis of no significant difference between the mean scores for the two groups was tested using the \( t \)-test. Independent sample \( t \)-test outcomes have a two-tailed significance \( (p\text{-value}) \) of 0.398, indicating that no significant difference existed between pretest scores of the two groups. Thus, we infer that students in the medium-achievement category in the two groups have similar computer skills.

### Table 3. Group statistics of pretest for the students in medium-achievement category

<table>
<thead>
<tr>
<th>Group</th>
<th>( N )</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>The experimental group</td>
<td>10</td>
<td>81</td>
<td>5.16</td>
<td>1.63</td>
</tr>
<tr>
<td>The control group</td>
<td>10</td>
<td>79</td>
<td>5.16</td>
<td>1.63</td>
</tr>
</tbody>
</table>

### Table 4. Independent sample \( t \)-test of pretest for the students in medium-achievement category

<table>
<thead>
<tr>
<th>( F )</th>
<th>Sig.</th>
<th>( t )</th>
<th>( df )</th>
<th>Sig. (two-tailed)</th>
<th>Mean Difference</th>
<th>95% CI Lower</th>
<th>95% CI Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.443</td>
<td>0.514</td>
<td>0.866</td>
<td>18</td>
<td>0.398</td>
<td>2</td>
<td>−2.85</td>
<td>6.85</td>
</tr>
</tbody>
</table>

CI = Confidence Interval, \( df \) = degrees of freedom

Table 5 lists the means and standard deviations for the nine students in low-achievement category in both groups, followed by the difference between means. The mean and standard deviation for the nine students in the experimental group were 58.88 and 12.19, respectively, while those for the nine students in the control group were 60.55 and 12.85, respectively. Table 6 shows \( F = 0.066 \) and Sig. \( p = 0.801 \), which indicates that \( p > 0.05 \) and equal variances can be assumed. Thus, the null hypothesis of no significant difference between mean scores for the two groups was tested using the \( t \)-test. Independent sample \( t \)-test outcomes show two-tailed significance \( (p\text{-value}) \) was 0.781, indicating that no significant difference exists between pretest scores of the two groups. Accordingly, we infer that students in the low-achievement category of the two groups have similar basic computer skills.

### Table 5. Group statistics of pretest for the students in low-achievement category

<table>
<thead>
<tr>
<th>Group</th>
<th>( N )</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>The experimental group</td>
<td>9</td>
<td>58.88</td>
<td>12.19</td>
<td>4.06</td>
</tr>
<tr>
<td>The control group</td>
<td>9</td>
<td>60.55</td>
<td>12.85</td>
<td>4.28</td>
</tr>
</tbody>
</table>

### Table 6. Independent sample \( t \)-test of pretest for the students in low-achievement category

<table>
<thead>
<tr>
<th>( F )</th>
<th>Sig.</th>
<th>( t )</th>
<th>( df )</th>
<th>Sig. (two-tailed)</th>
<th>Mean Difference</th>
<th>95% CI Lower</th>
<th>95% CI Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.066</td>
<td>0.801</td>
<td>−0.282</td>
<td>16</td>
<td>0.781</td>
<td>−1.66</td>
<td>−14.18</td>
<td>10.85</td>
</tr>
</tbody>
</table>

CI = Confidence Interval, \( df \) = degrees of freedom

Table 7 through 12 show \( t \)-test results of the post-test between the high-, medium-, and low-achievement categories in the experimental and control groups, respectively. The mean and standard deviation for the 10 students in the high-achievement category in the experimental group were 82 and 20.44, respectively, while those for the 10 students in the control group were 59 and 26.854, respectively, as given in Table 7.

Table 8 also exhibits \( F = 0.858 \) and Sig. \( p =0.367 \), demonstrating that \( p > 0.05 \) and equal variances can be assumed. Thus, the null hypothesis of no significant difference between the mean scores for the students in the high-achievement category in both groups was tested. It can be seen from Table 8 that the \( t \)-value was 2.155, the degrees of freedom were 18, the two-tailed significance \( (p\text{-value}) \) was 0.045, and the difference between two means was significant at \( p < 0.05 \). Therefore, we infer that mean scores in the high-achievement category in the experimental group and control group have obvious differences. The students in the high-achievement category in the experimental group improved their ability to retrieve and analyze information when guided by the learning-assistance module.

### Table 7. Group statistics of post-test for the students in high-achievement category

<table>
<thead>
<tr>
<th>Group</th>
<th>( N )</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>The experimental group</td>
<td>10</td>
<td>82.00</td>
<td>20.440</td>
<td>6.464</td>
</tr>
<tr>
<td>The control group</td>
<td>10</td>
<td>59.00</td>
<td>26.854</td>
<td>8.492</td>
</tr>
</tbody>
</table>
Table 8. Independent sample $t$-test of post-test for the students in high-achievement category

<table>
<thead>
<tr>
<th>$F$</th>
<th>Sig.</th>
<th>$t$</th>
<th>$df$</th>
<th>Sig. (2-tailed)</th>
<th>Mean Difference</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.858</td>
<td>0.367</td>
<td>2.155</td>
<td>18</td>
<td>0.045</td>
<td>23.000</td>
<td>0.579</td>
<td>45.421</td>
</tr>
</tbody>
</table>

CI = Confidence Interval, $df$ = degrees of freedom

Table 9 and 10 list the post-test statistics for students in the medium-achievement category. The mean and standard deviation for the 10 students in the medium-achievement category of the experimental group were 59 and 24.244, respectively, while those for students in the control group were 51 and 15.239, respectively, as shown in Table 9.

Table 10 shows $F = 5.038$ and Sig. $p = 0.038$, implying that $p < 0.05$ and unequal variances can be assumed. Thus, the null hypothesis of no significant difference between mean scores for students in the medium-achievement category in both groups was tested. Test results show that the $t$-value was 0.883, the degrees of freedom were 15.151, the two-tailed significance ($p$-value) was 0.391, and the difference between two means was insignificant at $p > 0.05$. That is, no obvious difference existed between the experimental group and control group in the medium-achievement category. Only a small number of students had satisfactory ICT application capabilities in the medium-achievement category; thus, their performance could be improved with assistance from the learning diagnostic module.

Table 9. Group statistics of post-test for the students in medium-achievement category

<table>
<thead>
<tr>
<th>Group</th>
<th>$N$</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental</td>
<td>10</td>
<td>59.00</td>
<td>24.244</td>
<td>7.667</td>
</tr>
<tr>
<td>Control group</td>
<td>10</td>
<td>51.00</td>
<td>15.239</td>
<td>4.819</td>
</tr>
</tbody>
</table>

Table 10. Independent sample $t$-test of post-test for the students in medium-achievement category

<table>
<thead>
<tr>
<th>$F$</th>
<th>Sig.</th>
<th>$t$</th>
<th>$df$</th>
<th>Sig. (2-tailed)</th>
<th>Mean Difference</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.038</td>
<td>0.038</td>
<td>0.883</td>
<td>15.151</td>
<td>0.391</td>
<td>8.000</td>
<td>-11.284</td>
<td>27.284</td>
</tr>
</tbody>
</table>

CI = Confidence Interval, $df$ = degrees of freedom

Table 11 and Table 12 list the post-test statistics for students in the low-achievement category. Table 11 lists the means and standard deviations. The mean and standard deviation for the nine students in the experimental group were 36.67 and 20.616, while those for the nine students in the control group were 41.11 and 20.883, respectively.

Table 12 shows that $F = 0.063$ and Sig. $p = 0.806$, suggesting that $p > 0.05$ and equal variances can be assumed. Thus, the null hypothesis of no significant difference between mean scores was tested. The test results show that the $t$-value was $-0.454$, the degrees of freedom were 16, and the two-tailed significance ($p$-value) was 0.656. We conclude that the difference between two means was insignificant at $p > 0.05$, and no obvious difference existed between the experimental group and control group in the low-achievement category.

Based on $t$-test results of the pretest, it was observed that students in each achievement category in the experimental group and control group had similar basic computer skills. The students in the high-achievement category in the experimental group performed significantly better when using the learning-assistance system because students in the high-achievement category typically had a strong ability to analyze information. The performance of most students in the medium- and low-achievement categories did not improve when the students were assisted by the learning module.
diagnostic module because these students lacked basic ICT application capabilities and had difficulty using the application software suite provided by the platform. We thus infer that the proposed learning-assistance tool proved helpful to students with basic computer skills.

Additionally, the teacher observed that students in elementary school are typically impatient with text information given by the assessment platform. We believe this phenomenon is strongly correlated with recent advancements in multimedia information technology. Students are accustomed to accessing multimedia information, such as audio, video, and still images in daily life, such that they tend to lack skill in concentrating and analyzing text information. By receiving timely feedback from the learning-assistance tool, students were guided to learn how to collect, analyze, and integrate information, and solve daily life problems using their computer skills.

To verify whether hints or feedback provided by the learning-assistance tool were appropriate, a short questionnaire was given to students in the experimental group that included the question “Is the learning system able to provide you with appropriate hints or feedback?” Among the 29 participants, 23 believed that the hints or suggestions were useful when they went in the wrong direction during the assigned projects. Furthermore, the teacher who participated in this study was asked to manually check whether the system offered useful and timely feedback that guided students when they encountered a problem. Table 13 lists the accuracy of feedback. Most feedback or hints were beneficial for students in determining the next step.

Table 14 shows the number of feedback messages at three levels provided to students in the three achievement categories when they encountered a problem. Students in the high-achievement category use software frequently, and most completed the mission with only weak hints provided by the system. Conversely, most students in the low- and medium-achievement categories needed moderate and strong hints because they were unfamiliar with using the software. However, they still had difficulty using the software after receiving the hints or suggestions as a result of their poor ICT application capabilities. Accordingly, these students needed to spend considerable time making progress during the assigned projects. Therefore, their post-test outcomes were often poor because they failed to complete the assigned projects before the end of the assessment period, although most students believed that the hints or suggestions provided by the system were useful. In order to help the students in the low- and medium-achievement categories effectively when they use the proposed learning-assistance platform, teachers need to spend more time to help students be familiar with using application software suite provided by the platform prior to participating in the assessment. Meanwhile, the scale of the project can be reduced to allow the students in the low- and medium-achievement categories to have enough time to complete the assigned projects at the end of the assessment.

Table 13. Accuracy of feedback provided by the system

<table>
<thead>
<tr>
<th>Feedback</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>203</td>
<td>29</td>
<td>232</td>
</tr>
<tr>
<td>Percentage</td>
<td>87.5%</td>
<td>12.5%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 14. The number of feedback messages in three levels provided for the students in three achievement categories

<table>
<thead>
<tr>
<th>Number of feedback messages in three levels</th>
<th>Weak hint</th>
<th>Moderate hint</th>
<th>Strong hint</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-achievement category</td>
<td>13</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Medium-achievement category</td>
<td>4</td>
<td>16</td>
<td>30</td>
</tr>
<tr>
<td>Low-achievement category</td>
<td>2</td>
<td>40</td>
<td>124</td>
</tr>
</tbody>
</table>

Conclusions and future work

Project-based activities were incorporated into a novel assessment platform to meet the pressing needs of primary school students in Taiwan. Students were expected to apply application software suite provided by the platform to solve daily life problems. The assessment module evaluated student learning outcomes using learning portfolios and provided the teacher and students with final reports. A well-known machine-learning technique, reinforcement learning, was used to track learner operations and provide suggestions to learners when they were confused or stalled when using the ICT literacy assessment platform. To verify the effectiveness of the reinforcement learning
mechanism in the learning-assistance platform, 58 Grade 5 students from two classes participated in the experiment. Experimental results reveal that students in the high-achievement category in the experimental group performed better after using the learning-assistance system. Most students in the low- and medium-achievement categories had difficulty using the software due to their poor ICT application capability. Accordingly, they spent additional time making progress during the continuous assigned projects. Therefore, their post-test outcomes were often poor because they failed to complete the assigned projects before the end of the assessment period. Furthermore, teaching loads were significantly reduced because appropriate hints or feedback were automatically provided to learners without teacher involvement. In future work, student learning portfolios will be used and additional appropriate machine-learning techniques will be adopted to establish a probability model for each action node on the path toward completion of the assigned task and to automatically determine the reward value at each node, which is currently set by domain experts.

Acknowledgments

This research was partially supported by Ministry of Education and National Science Council of Taiwan (Contract Numbers NSC 96-2628-E-259-022-MY3 and NSC 98-2220-E-259-003). Ted Knoy is appreciated for his editorial assistance.

References


