J. Comput. Educ.
https://doi.org/10.1007/s40692-020-00162-9

Effects of a peer tutor recommender system (PTRS) with machine learning and automated assessment on vocational high school students' computer application operating skills

Zhao-Heng Ma[1,2] · Wu-Yuin Hwang[3] · Timothy K. Shih[1]

Received: 28 October 2019 / Revised: 16 March 2020 / Accepted: 19 March 2020

Abstract Many information technology courses frequently use the learning by doing strategy in vocational high schools. Particularly, learning computer application operating skills is essential for students because excellent computer application operating skills can help them attain good jobs. However, when fostering students' computer application operating skills by teaching in vocational high school using the learning by doing strategy, a teacher learns that helping all students, evaluating their learning problems, and providing feedback to correct their mistakes are challenging. After investigating the challenge, a machine-learning-based peer tutor recommender system (MPTRS) with automated assessment was proposed to enhance students' learning performance in computer application operating skills. The advanced automated assessment system (AAS) used computer vision technology to evaluate student assignments and instantly return feedback. The recommendation mechanism of the MPTRS enhanced mutual help among students based on their social relationships, learning performance, and recommendation feedback. Furthermore, machine-learning techniques were used to improve recommendations. In the experiment, the experimental group used the proposed system, and the control group used a

✉ Zhao-Heng Ma
maxma1021@gmail.com

Wu-Yuin Hwang
wyhwang@cc.ncu.edu.tw

Timothy K. Shih
timothykshih@gmail.com

[1] Department of Computer Science & Information Engineering, National Central University, No. 300, Zhongda Rd., Zhongli Dist., Taoyuan 32001, Taiwan, ROC

[2] Taoyuan Municipal Zhongli Home Economics and Commercial Senior High School, No. 36, Deyu Rd., Zhongli Dist., Taoyuan 32041, Taiwan, ROC

[3] Graduate Institute of Network Learning Technology, National Central University, No. 300, Zhongda Rd., Zhongli Dist., Taoyuan 32001, Taiwan, ROC

Published online: 02 April 2020

conventional commercial automated grading system. From the experimental results, the learning performance of the experimental group significantly improved between the pretest and post-test. Students can correct and complete more assignments using the advanced AAS and students who behind in learning also can use the peer tutor recommender function for asking help. Participants were also satisfied with the proposed advanced AAS and MPTRS. It is worth to promote the proposed system to teachers of adopting the learning-by-doing strategy in computer application classes.

Keywords Learning by doing · Peer tutoring · Recommender system · Automated assessment system · Machine learning · Computer vision

Introduction

Teachers in subjects such as computer application skills, mathematics, physics, chemistry, and computer accounting often allocate course time for teaching and practice in the classroom. This is the concept of learning by doing (Harger 1996). Properly assisting students in practice is the topic of many studies.

In the age of information technology (IT), the global demand for mastering IT skills is immense. Accordingly, vocational high schools provide an introductory computer course (ICC) to enhance students' IT skills (Gibbs et al. 2015). ICCs primarily focuses on computer application operating skills such as word processors, spreadsheets, presentations, and accounting software. Computer application operating skills courses most frequently use learning by doing strategy.

The conventional method of teaching productivity software applications in vocational high schools consists of two phases: the teaching phase and the practical and assessment phase. In the teaching phase, a teacher demonstrates a sequence of operation steps using a screen broadcast system in the computer classroom. In the practical and assessment phase, students must complete short assignments to ensure their familiarity with the applications.

Students typically experience challenging situations throughout their learning process (Boud et al. 2014; Damon and Phelps 1989). In practical sessions, if students have difficulties learning, then they can ask their classmates or raise their hands to ask the teacher. However, a teacher attempting to solve a few students' difficulties during practical sessions may exceed the time scheduled for that phase and may affect teaching progress. Moreover, a teacher may feel exhausted by providing individual guidance to more than two students during a practical session. Chao (2007) observed that students experience learning difficulties in computer classes. Most students requested help from peers (73.4%), followed by teachers (43.8%) and searching for information or reading books (38.6%). Therefore, teachers require a strategy to help students efficiently.

Damon and Phelps (1989) and Goodlad and Hirst (1989) proposed that peer learning can be divided into peer tutoring, cooperative learning, and peer collaboration, with conditions of reciprocity between peers. In peer tutoring, students are divided into experts and novices who use the same learning materials, and this learning strategy, which is characterized by a high level of active cooperation, benefits both tutors and tutees. The tutor performs the role of the teacher, and the tutees comprise students with low learning performance (Conrad 1974). Peer tutoring is the most popular strategy and is widely used in different education fields. According to Topping and Ehly (1998), the learning outcomes of both the tutee and tutor can be improved. Although the teaching quality provided by peers may not be professional, peer tutoring can provide one-on-one assistance to solve the tutee's problems in real-time. This process can help the tutee focus more on the learning process, which cannot be achieved in traditional teacher instruction (Rohrbeck et al. 2003).

Researchers have examined recent research findings related to peer tutoring approaches and have recognized that in conventional fixed-role strategies students are grouped statically by instructors (Duran 2017; Good and Lavigne 2017).

In static grouping strategies, instructors must evaluate student abilities before class, but dynamic grouping is based on students' current learning performance. As a result, the performance of both strategies has the same excellent effect. In the static grouping strategy, sometimes it is not easy to evaluate student abilities before class. In the absence of pretest results, the dynamic grouping is a good strategy. Lee and Hu (2018) designed a system that used a dynamic grouping strategy to assign tutors with considering assignment submission time and the frequency of assisting others.

Familiarity and friendship are essential factors in help-seeking and classroom social environment likely influences students' help-seeking behavior (Ryan et al. 1997; Aronson et al. 2016). However, grouping by instructors often does not take into account the student's social relationship.

Therefore, developing a system that uses the peer tutoring strategy to select appropriate tutors necessitates solving the problem of student assignment marking. In the conventional assessment process, the teacher marks student assignments and provides feedback. However, individually evaluating students in a computer application class is time intensive and may increase the teacher's workload. This task is performed separately, which means that students' learning situations cannot be understood immediately.

Teachers' evaluation of a large number of students' assignments and provide instant feedback is very time-consuming. Acquiring an automated assignment marking solution will help minimize teachers' additional workload of the computer application class. As a result, e-assessments and automated grading systems (AGSs) have been developed to automatically mark and correct assignments. Automated assessment systems (AASs) provide detailed feedback for every question and reasoning for incorrect answers. Therefore, these systems improve student learning. This process could reduce teacher workload and simplify interactions between teachers and students (Lahtonen and Isomöttönen 2012). AGSs were developed for targeted certification examinations. These systems significantly enhance the learning process and efficiently manage the grading process while providing a total assessment (Matthews et al. 2012). Information and communication technology certifications, such as Microsoft Office User Specialist (MOS), Cambridge International IT Skills, and Techficiency Quotient Certification (TQC) of the Computer Skills Foundation (CSF) have become popular (Computer Skill Foundation 2019). AGSs can be installed only in specific examination centers, and teachers cannot use them while teaching. The TQC is familiar in Taiwan, and CSF has provided a limited version of the TQC for teachers to use their automated marking function in computer application courses. Although the limited version can mark student answers automatically, but it cannot record the learning process or leave comments for student errors.

However, because the technology used in AGSs involves simulation or file comparison (Matthews et al. 2012), it is often limited to specific operations, and the same result obtained using different operations is not accepted. Besides, if the back end of the file is not stored as XML or other open formats, it is difficult to be used by the above method.

The performance of image processing libraries related to computer vision has rapidly improved in capturing and recognizing objects and images. Sinha et al. (2019) and Riad et al. (2017) developed a module-training algorithm that does not involve data labeling, and analyzes the image to extract information in the document. Therefore, using computer vision, different operations are acceptable when the output results are the same in computer applications. Using image processing to evaluate Word, Excel, PowerPoint, and other application software in web design, accounting, and statistical analysis is unlimited.

On the basis of the aforementioned observations, in this study, a machine-learning-based peer tutor recommender system (MPTRS) with automated assessment was developed to enhance student-learning performance. The proposed system marks assignments and automatically recommends peer tutors by considering students' performance and relationships. The back end of the advanced AAS involves computer vision technology that flexibly evaluates student assignments and instantly returns feedback. The MPTRS involves collaborative learning activities, including a peer tutor recommender, by considering the students' social relationships, computer application operating skill-learning performance, and recommendation results in feedback as training data. To accomplish the objectives of this approach, the back end of the system involves modules for data extraction, data analysis, and recommendation techniques using machine learning. The proposed system was evaluated through comparisons between a control group (CG) and an experimental group (EG) of students. The CG used automated marking with the limited version of a commercial AGS for learning activities, and the EG used the learning activities of the proposed system.

The MPTRS system automatically recommends tutors on the basis of students' learning performance and adjusts tutor list using student feedback. First, the system requires students to fill the questionnaire before starting the class. The questionnaire designed to identify social relationships. We are performed by the system, which can considerably reduce teacher workload. Various analyses and evaluations were performed to investigate the following research questions:

- Does this system improve computer class learning outcomes?
- Does this system recommend tutors with high learning performance?
- Does this system consider social relationships among students when recommending tutors?
- Are students satisfied with the advanced AAS?
- Are students satisfied with the MPTRS?

Related works

Peer tutoring strategy

Peer tutoring is a systematic approach of learners helping each other and learning through teaching (Good and Lavigne 2017; Goodlad and Hirst 1989; Falchikov 2001) has a positive effect on increasing academic performance (Lacaba et al. 2018). This strategy can be practiced in various combinations: students to students, non-professional adults to adults, and children to children. Topping (2000) presented a broad definition of peer tutoring that "involves people from similar social groupings who are not professional teachers helping each other to learn and learning themselves by teaching." These activities are organized and monitored by teachers in the classroom. Peer tutoring is not only a viable and effective method of improving the learning performance of tutors and tutees but also an opportunity for teachers to evaluate the peer-tutoring process in class. Boud et al. (2014) argued that peer learning is mutually beneficial when students share their knowledge, experiences, and ideas with others. During the peer tutoring process, highly gifted students can help others who have limited abilities, and teachers can coordinate this interaction. Because peers may have common interests, using relevant relationships helps both participants assimilate ideas effectively. Peer tutoring can prevent failure, lead to assignment success, and improve the possibility of long-term education and independent learning (Ryan et al. 1997). It is beneficial for not only the tutee but also the tutor. Tutors have the opportunity to clarify and elaborate their understanding by explaining learning content to tutees (Falchikov 2001). Chuang et al. (2018) also found that students learning with the peer-tutoring strategy had significantly higher learning achievement and concept-mapping scores than students of conventional computerized concept-mapping approach.

Classwide peer tutoring

The classwide peer tutoring (CWPT) strategy for use in various educational systems (Greenwood et al. 1989; Hawkins et al. 2019). In CWPT, the classroom is an activity space, and teachers must organize teaching materials before class and conduct quiz competitions during class. During the game, students are divided into two groups. Students in each group are paired. By answering questions on flashcards, students alternate between the tutor and tutee roles. The tutor is responsible for correcting the tutee's answer and awarding a score. The roles reverse after a round. Teachers are responsible for competition scoring and event management. Compared with the conventional model, this model can improve spelling (Greenwood and Delquadri 1995), reading, and mathematics (Greenwood et al. 1989). Because CWPT does not have a fixed grouping strategy, low-achieving students or those with learning disabilities are considerably limited in their participation and their learning process.

Greenwood et al. (2001) improved peer interaction and guidance by employing heterogeneous grouping to make CWPT more structured with peer-assisted learning strategies (PALS). Similar to CWPT, students are paired; one is the tutor, and the other is the tutee. After completing an activity, the roles reverse. The teacher must select high-ability students before class. After training, they assist and teach students with lower abilities. By ranking abilities from high to low, students are divided into two segments. Students with the highest scores in the first segment are paired with those with the highest scores in the second segment. During the activity, the two roles exchange continually. The students with high abilities should be demonstrated first, and students with lower abilities can emulate.

Reciprocal peer tutoring

Reciprocal peer tutoring involves two students teaching each other during the tutoring interaction and does not explicitly define individual tutor and tutee roles. Students ask and answer questions and correct and confirm answers. After completing a learning activity, the two roles exchange. This strategy uses mutual help and homework discussion to share learning experiences and accomplish tasks (Fantuzzo et al. 1995). This differs from PALS because the tutor does not score the tutee. Finally, an individual assessment is adopted, but students cannot discuss the test or assist each other. Ketele et al. (2010) argued that when student ability levels are significantly different without considering social relationships, interactive activities suffer, particularly when the tutor is incompetent. Therefore, peer-tutoring activities must incorporate methods to ensure a match between tutors and tutees.

Fixed-role peer tutoring

Fixed-role peer tutoring involves initially organizing a group of tutors and tutees with no opportunity to exchange roles (McKellar 1986). In this method, students with experience or high learning ability help students without experience or with a low ability to learn new knowledge and skills. Generally, this model is based on a one-on-one teaching process and can be used to help students with individual needs develop skills and enhance confidence. Alegre et al. (2019) also found that fixed and same-age peer tutoring may be very beneficial in middle school algebra classes. These tutors and tutees are structured peers. The interactive approach can increase opportunities for students to interact, which improves their interpersonal communication skills and friendship. Topping and Ehly (1998) highlighted that tutor capabilities are crucial in fixed-role peer-tutoring interaction. If the tutor delivers inaccurate information or is unprepared, the tutee responds negatively, and their performance declines.

Social relationships in classroom

Ryan et al. (1997) discussed many academic situations in which students do not ask for assistance when required. Even though students have difficulties in providing answers for assignment questions, some students avoid asking for help from their peers for various reasons (Aronson et al. 2016). They manage to answer questions based on their knowledge, but their answers are sometimes incorrect. The social relationship among peers is an essential reason for the reluctance to ask for help. Ryan et al. (1997) argued that familiarity and friendship are essential factors in help seeking. Therefore, classroom social environment likely influences students' help-seeking behavior (Aronson et al. 2016). In such situations, students feel uncomfortable asking for help from their peers, and this discomfort may affect their academic performance. Unfamiliarity with peers prevents students from giving and seeking help and may negatively affect learning (Dong et al. 2018). Madaio et al. (2018) also found that high-rapport dyads peer tutors provide more help to tutee to explain the reasons than low-rapport dyads peer tutors.

Computer-aided system

Evans and Moore (2013) explored and applied computer-aided mentoring using an online peer-assisted learning electronic peer coaching framework known as Online Peer-Assisted Learning (OPAL). When students answer a question correctly, they can access the "pool" of tutors. Student can then anonymously post a "tutoring ticket" in the platform database to indicate that they are eligible to teach the question or problem. This process allows tutees to select qualified tutors. The tutoring ticket contains the history of tutors' instruction and previous tutoring session length. The process concludes when the student answers the question and is granted access to the tutor pool.

Akobe et al. (2019) proposed a web framework for online peer tutoring applications. This application identifies two key target users: the tutor and tutee. The tutor is responsible for accepting requests from tutees and teaching the students in the tutoring session. They are also responsible for uploading essential documents to the platform for tutees to access. In addition, the tutees can ask for tutoring sessions and search for tutors when required (Akobe et al. 2019). This method can confirm the tutors' abilities because students who access the tutor pool have completed their work. Because tutors also provide teaching materials, tutees can select an appropriate tutor.

However, face-to-face tutoring in the computer classroom still requires improvement because tutors cannot anonymous. Lee and Hu (2018) used a dynamic grouping strategy for peer tutoring in computer classrooms. They provided system records the number of times a student submits an assignment and the number of other tutors. When students press the help button, they become tutees and enter a queue to wait for a tutor. When students complete and upload the assignment to the system, they will be assigned to help tutees with learning difficulties face to face. This mechanism considers student performance and the number of other tutors before selecting an appropriate tutor.

Recommender systems

Recommender systems that use machine learning are broadly used to recommend services or content such as products, jobs, majors, courses, books, learning content, citations, films, and music to end users by observing their interests (Linden et al. 2003; Lu et al. 2015; Ochirbat et al. 2018). The recommender system of the popular e-commerce website Amazon suggests products based on user purchase or search patterns (Linden et al. 2003). LinkedIn recommends jobs, groups, or companies that might be of interest to professionals, experts, colleagues, or students. Facebook suggests people to become friends with or groups that users might want to follow. User profiles contain two types of data: interaction data and social connection data (Diaby et al. 2013). On the basis of the study of Diaby and Viennet (2014), personal information or user behavior from social media can be used to create a recommender system or to match people.

Machine learning methods

Machine learning methods have achieved great success in various fields. As a result, researchers have started to explore deep learning techniques in the field of recommendation (Wang et al. 2017; Zhang et al. 2017).

Designing a recommender system involves addressing challenges of accurateness, scalability, and confidentiality. Conventional recommender systems are based on the most commonly used algorithms of content-based (CB) filtering, collaborative filtering (CF), and hybrid filtering (Lu et al. 2015). CB filtering produces recommendations for active users according to similarities between items. CF-based recommendation techniques help people make selections based on the opinions of others who share similar interests. In user-based CF approach, a user receives recommendations for items linked by similar users. In item-based CF, a user receives recommendations for items that are similar to those they enjoyed in the past (Lu et al. 2015; Zhang et al. 2014). Hybrid recommendation combines the most useful features of two or more recommendation techniques to achieve higher performance and overcome the drawbacks of traditional recommendation methods (Ochirbat et al. 2018; Burke 2002).

Recommender system in education

Recommender systems are also frequently used in education environments. Xie et al. (2019) analyzes the publications in the journal, personalized recommendation is one learning support for the learning process of adaptive/personalized systems. Klašnja-Milićević et al. (2011) developed a recommendation module of a programming tutoring system that used to automatically adapt to the interests and knowledge levels of learners. Ochirbat et al. (2018) collects three sets of information including student’s profiles, vocational interests, and their behaviors to recommends occupation by integrated content-based collaborative filtering methods. Zaíane (2002) proposed an agent for e-learning systems to recommend learner actions based on previous learner actions in an e-learning context. Mihaescu et al. (2015) used a machine learning classification procedure to retrieve the most suitable tutors in online education environments. Letting learners find the most suitable tutors can help them online.

Automated grading systems

The IT skill levels of vocational high school students in an ICC differ because they enroll in the course from different junior high schools. If students identify where they have answered incorrectly and can find answers on their own by giving hints, their learning performance can increase (Matthews et al. 2012; Chu et al. 2019). Increasing the number of problems, assignments, and exercises can also reinforce student learning (Murray 1998). However, providing feedback for student assignments in large classes is difficult. Reasons for selecting AGSs and e-assessment systems include cost-effective grading (Swithenby 2006), rapid feedback in the form of marks and comments for students, and objective and consistent assignment marking. Therefore, students can frequently review their assignments for improving the learning performance of students in a personalized learning environment. (Whitelock and Brasher 2006; Wongwatkit et al. 2017).

AGSs are categorized into two types (Matthews et al. 2012). The first type uses procedural-based grading, which is performed by the application simulator. These simulators respond to the procedure of students' responses (keystrokes and mouse clicks) to complete a task. The second type provides instructions for formatting and modifying a blank or original document. Students upload the completed file, and the document is graded. The system provides instant feedback and based on errors.

Automated assessment technologies

Automated assessment technologies developed to analyze student-produced documents are classified into four categories: (1) visual basic for applications (VBA), (2) component object model (COM), (3) extensible markup language (XML) technology, and (4) artificial intelligence (AI) and other techniques (Chorana et al. 2015).

First, VBA technology is a version of the Visual Basic language included in MS Office. VBA runs in the background and responds to events such as opening a form or clicking a command button. Most e-assessment systems have been developed on the basis of VBA technology to easily communicate with MS Office programs (Koike et al. 2007; Ren et al. 2010). In e-assessment systems for MS Word, instructors establish grading criteria, such as page settings, paragraphs, fonts, colors, text, indents, figures, and tables that the systems use to mark student files. Moreover, the system can be used to verify whether students have correctly used each feature of MS Word. Students receive feedback in messages. Because VBA is a part of MS Office, the systems require identification for each question. Therefore, this category of systems cannot flexibly modify or add new questions.

The second category of e-assessment systems uses COM technology, which is a binary interface standard developed by Microsoft. The scoring criteria information for required IT skills is recorded in a database table. The e-assessment system extracts attributes of the student file and uses the COM interface to match them with the scoring information (Zhu and Shen 2013).

Regarding the third strategy, Lahtonen and Isomöttönen (2012) used XML technology to evaluate stylistic and technical correctness of office files. The scoring criteria file representing the requirements for an IT skills assignment and containing required style information for checkable items, such as bold and page numbering, was configured using XML. Chorana et al. (2015) provided a simplified fully automated assessment approach with a standard representation using XML. Similarity between the teacher's correct and student's produced documents was measured using an XML tree representation. The fourth category of applications based on AI technology can evaluate the rich text format file output of any word processor to compare the students' produced files and the teacher's correct solution. The application then categorizes errors by type and generates a report (Long et al. 2003).

The other strategy is using "combine and compare." MS Word allows systems to merge documents to identify differences between them using this function. Thus, the student's produced document merges with the instructor's correct version, and the differences are then recorded in an MS Office Access table. Although this method is compelling, it is limited to MS Office, and some skills (e.g., manipulating text boxes) are difficult to evaluate (Hill 2011).

Computer vision technology

Sinha et al. (2019) and Riad et al. (2017) use contour detection methods of computer vision to locate the table and Optical Character Recognition (OCR) for text extraction and regular expression for string comparison. These kinds of documents are not produced by Microsoft applications and are not stored in XML or other open formats. With the improvement of computer vision technology, related applications are also increasing. In recent years, the open source computer vision library (OpenCV 2019), an open source computer vision and machine learning software library, has become increasingly used in commerce and research(Bradski and Kaehler 2008; Puri and Jain 2019). OpenCV provides a common infrastructure for computer vision applications. The library has more than 2500 optimized algorithms, including a complete set of classic and modern computer vision and machine learning algorithms. These algorithms can be used to identify objects and search for similar images from image databases. It has C, Python, Java, and MATLAB interfaces and supports Windows, Macintosh, Linux, and Android operating systems. OpenCV is mainly used for real-time vision applications and uses MMX and streaming single-instruction, multiple data extension instructions. For OpenCV to work efficiently with Python 3.6, the NumPy and Aircv packages must first be installed (OpenCV).

We proposed a novel strategy for automated assessment marking using computer vision technology. The instructor's correct documents are recorded as scoring criteria images. The student's work is extracted from the work area of their screen as answer images. We define marking information such as evaluating information and write comments to incorrect answers. The proposed system compares the student's answer with the criteria on the basis of computer vision technology and provides feedback with scores and comments for students. The system then records the students' learning progress and uses it in the next stage to recommending tutors.

System design and implementation

An MPTRS that combines machine learning techniques with an automated assessment approach was designed and developed (Fig. 1). The MPTRS is a Web application running on any web browser on a personal computer (PC) or mobile device. First, the students complete their assignments and upload them to the MPTRS using a web browser. The system then marks the student-produced document and provides feedback in the form of scores and comments on the assignment. If students have difficulties, they can use the peer tutor recommender module. The goal of the MPTRS is to recommend an appropriate peer tutor using a machine learning algorithm and improve the students' learning outcomes.

The back end of the proposed system uses the Flask architecture to manage communication between the front end and the database and is written in Python. The online marking component uses HTML5 technology that supports the browser to paste items from the Windows clipboard. In the system, students are not required to upload any produced files; they are required to only press the print screen button and paste the image in the web application to perform the marking process. The system identifies incorrect sections of the students' work and provides comments on how to correct the incorrect sections. Moreover, the system records the learning process. Students' assignment scores and peer tutor selection records are uploaded and recorded in Google Sheets. Teachers can then easily review the students' status by using Google Drive in the browser.

Advanced automated assessment module

In online automated assessment for a computer application operating skills class, marking students' completed assignments and monitoring their progress are challenging. Students may not be able to effectively evaluate their progress through a self-reported questionnaire. (Dong and Hwang 2012). Several AASs have been developed using different strategies, but they typically operate for necessary word processing skills such as document, character, and paragraph formatting (Chorana et al. 2015).

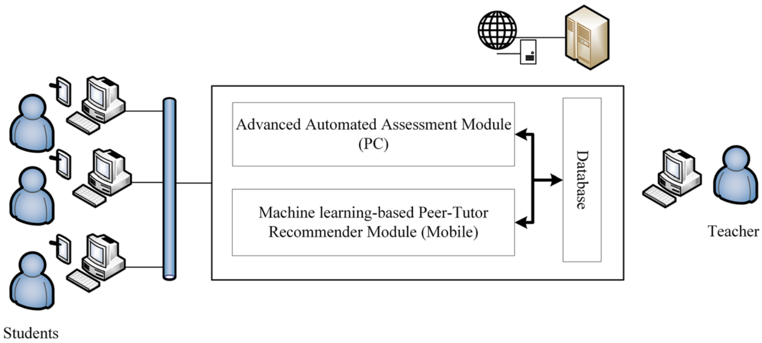

Fig. 1 Overview of functionalities used in the back end of the Web application system

The scoring criteria base on the instructor's correct document, and the answer image was extracted from the screenshot of the student's work. The system then loads the marking information such as evaluating section coordinates, matching confidence threshold information, and comments for incorrect answers from the Google sheet (Table 1).

After completing an assignment, the student changes the work area of Word document to a single-page mode and clicks the print screen button on the keyboard to capture the screen. The student then pastes (CTRL + V) the copied screenshot file in the WEB application to upload the file to the server. The system modifies both the student answer image and criteria image to the same size and cuts the criteria image into many small feature images. The system then uses computer vision technology to compare and match the student answer images and feature images. The matching confidence threshold was adjusted to increase matching accuracy and stability. If the threshold is too high, the system returns an error even when a student completes an assignment correctly. Moreover, if the matching threshold is too low, the system returns a correct answer even if the student completed the assignment incorrectly. After matching the feature images with the student's answer image, the system produces feedback results and comments.

Correct answer sections are marked in green circles, and incorrect sections are marked in red squares with comments, as illustrated in Fig. 2. The comments are provided in a Google sheet, where instructors can provide students with instructions. In studies, correctly marking documents that students create using different methods is difficult, particularly when documents include graphics or images, because two documents may be produced with the same layout but saved in different parameters. The current other AASs cannot resolve this challenge. In this advanced AAS, students can use different means of completing their produced files and assignments in the system. Students' records are stored in Google Sheets. Teachers can easily evaluate a student's assignments, understand their completion, and decide whether to proceed to the next course. The scoring results are also used as input data for the tutor recommender module. Moreover, students can continue submitting their produced files until they are entirely correct. The students' working screens are stored on the server, and the teacher can access them quickly. Student learning progress is then recorded, and the information is used in the next stage for recommending tutors.

Peer tutor recommender module

A neural network comprises numerous neurons and simulates a biological neural structure. After connecting artificial neurons and obtaining information, it can produce an output result by imitating the characteristics of biological nerves, repeating operations, and learning from experience. These output data can be further used (Specht 1991). Keras is a popular deep learning library widely used in AI applications. It is also a high-level neural network application programming interface (API) written in Python and can run with a TensorFlow back end (Day and Lin 2017; Gulli and Pal 2017; Nassar et al. 2020). Neural networks are classified by learning strategy and can be divided into supervised, unsupervised, and

Table 1 Sample representation of scoring criteria and comments for assignments

The checking section coordinate and size				Confidence threshold	Comments and instructions for incorrect answers in student documents
X axis	Y axis	Width	High		
464	700	75	59	0.95	Please review the line style of the page border
50	130	60	40	0.8	Please review “Fill,” “Patterns,” and “Apply to” under “Shading” in Borders and Shading
50	475	450	240	0.8	Please review “Alignment,” “Indentation,” and “Spacing” under “Indents and Spacing” in Paragraph
220	194	30	123	0.7	Please review the Chinese style setting
120	45	80	62	0.9	Please review whether the left and right indentations are eight characters. The border style uses three vertical lines and is three points wide

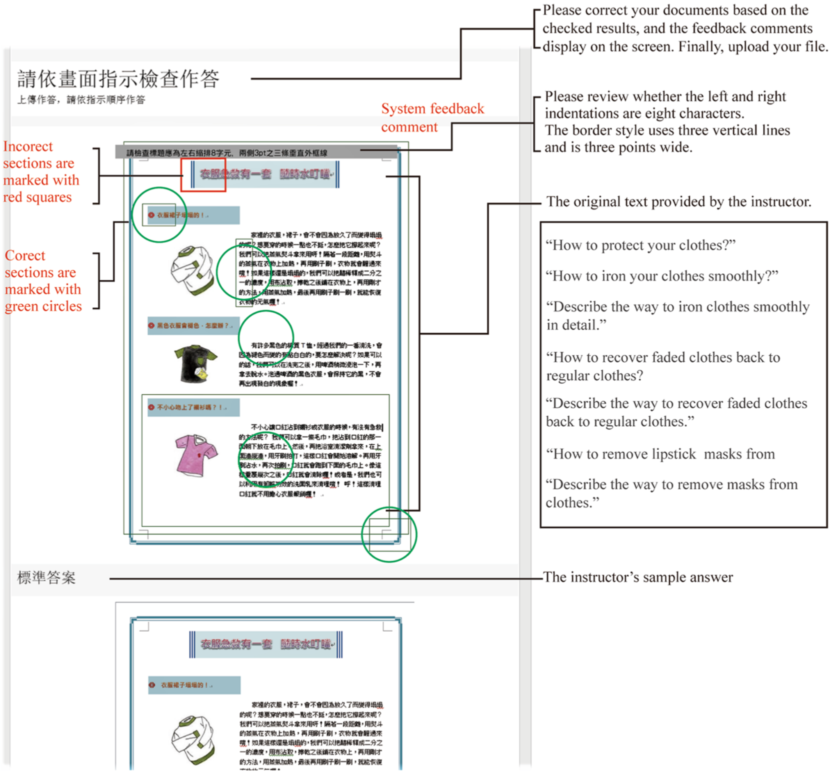

Fig. 2 Sample corrected result obtained from the advanced automated assessment system interface

associated learning networks (Sarkar et al. 2018). Machine learning methods such as matrix and tensor factorization have been frequently used in recommender system deep learning methods. Zhang et al. (2018) introduced a recommendation model based on deep neural networks that does not require additional information apart from interactions between users and items. First, they used a user-item rating matrix to obtain user and item features. These features were then used as the neural network input, and the output layer represented the probabilities of scores that the user might provide. Shepitsen et al. (2008) proposed a personalized recommender system that uses a social tagging system based on a hierarchical clustering method. Rahutomo et al. (2019) used Kears to develop a book recommender system based on user-generated ratings or known ratings as the main raw material to learn the pattern of favorable contents of each user. This study used Keras to develop a supervised learning network. The deep learning framework can easily create neural network embeddings as well as work with multiple input and output layers.

When developing the MPTRS, the Keras deep learning library was used to develop the neural network model with model-based collaborative filtering, which combines the student questionnaire, student performance, and recommendation

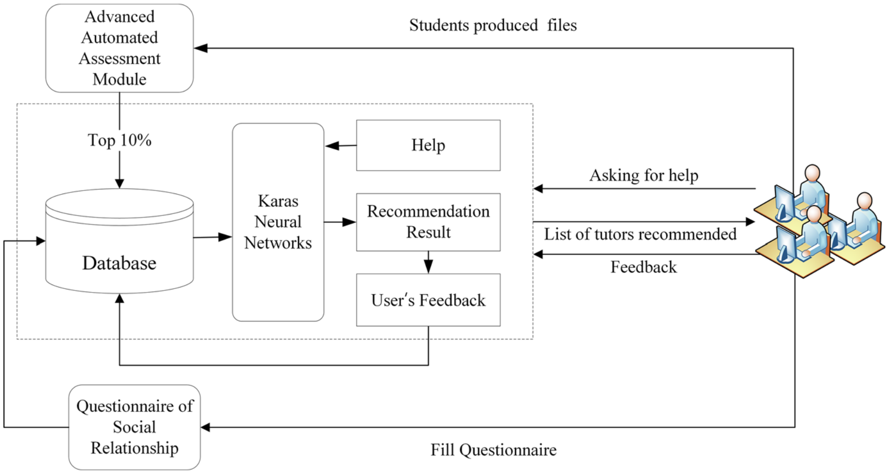

Fig. 3 System overview of machine learning-based peer tutor recommender module

feedback as the tutee-tutor raking matrix for input data (Fig. 3). The Keras Neural Network library (back-end engine is TensorFlow) was used in the proposed system, and the input layer, hidden layer, and output layer were fixed.

The model was then trained to generate the peer tutor priority order. When a student clicks the help button, the module returns a list of recommended peer tutors. The students can select a peer tutor and provide feedback for the selected peer tutor on the system. The peer tutor feedback form consists of five points (5–1): 5 (excellent): good computer application operating skills and familiar with the tutee, 4 (good): moderate computer application operating skills and familiar with the tutee, 3 (good): good with computer application operating skills but unfamiliar with the tutee, 2 (fair): moderate computer application operating skills but unfamiliar with the tutee, and 1 (poor): none of the candidate tutors are suitable for the tutee (Fig. 4). After completing the peer tutor feedback form, the tutee must inform the peer tutor and ask for assistance. The system also sends an email with the teacher's signature to remind the peer tutor. All students carry a mobile phone that notifies them when receiving an email. The system also records the feedback information.

The proposed model was constructed according to the following structure:

- Input: input tutor ID, tutee ID, and ranking data.

  1. Student questionnaire.
  2. The top 10% of students who completed the assignment are recorded, and the system generates a set of ranked records by location.
  3. Feedback is received from the peer tutor recommender result.

- Embedding layers: embeddings for tutors and tutees.
- Dot: combine embeddings using a dot product.

Fig. 4 Sample corrected result obtained from the MPTRS interface

In an embedding model, the embeddings are the weights learned during training. These embeddings can not only be used for extracting information about the data but also be extracted and visualized. Because of the work requirements, the proposed model did not use any fully connected layers. Two input layers were used during training: one for tutor ID and one for tutee ID. When implementing the recommendation model, tutors, tutees, and rank were used as input data. Peer tutors with the highest predicted ratings by students asking for help were then selected.

Although the marking module can run on a PC browser, the tutor recommender module is more suited for use on mobile devices because of privacy considerations.

Methodology

Creswell (2008) used a quasi-experiment. Accordingly, in this study, two classes participated in an experimental group (EG) and a control group (CG). During the practical and assessment phase, the EG used the system to mark their assignments online and recommended peer tutors for students who have difficulties learning. The CG used an automated marking system (AMS) in the limited version of TQC for assignment marking and individually sought help from classmates or the teacher. An independent-sample t test was used to analyze each group's pretest and posttest to determine whether learning performance differed. A paired-sample t test was also used to analyze whether the two groups differed before and after the experiment. The Pearson correlation measured the strength of the linear relationship between variables of posttest, number of assignments uploaded in class, frequency of asking for help, and frequency of offering to help others. Analysis of variance (ANOVA) was used to analyze differences between groups in students' roles in the class. The study was conducted at a public vocational school in Taiwan. The participant and instrument information is as follows: participants: 10th-grade students, EG = 38 (female = 35, male = 3) and CG = 36 (female = 34, male = 2), age: 15–16 years, major: home economics, instruments: pretest, weekly activity (word processing skills in computer application software), posttest, questionnaire, 2 h/week. The experimental schedule is displayed in Fig. 5.

In the first week, both the EG and CG completed a pretest. All students in the EG completed a questionnaire to select five students who they have social relationships with to ask for help when required. Students were introduced to using the advanced AAS. The word processing class was conducted from the second to sixth week. Both groups used the TQC Word 2010 textbook published by the CSF in Taiwan. In the teaching phase, the teacher conducted lectures for both groups. In the practical and assessment phase of the class from the second to sixth week, the CG used automated marking in the limited version of TQC and asked the teacher or peers for help face-to-face. The EG used the advanced AAS that marked assignments automatically and recorded the students' learning performance for the MPTRS. Because the MPTRS requires a large amount of data for training before applying machine learning techniques, for the first 3 weeks, students were not allowed to use the tutor recommend module. The MPTRS system used learning performance in student records as input data. Students started using the tutor recommender module to ask for help in the fourth week. In the EG, if the teacher discovered students who were behind in learning, the teacher advised them to use the tutor recommender function to ask for help. After the seventh week, the two groups conducted posttests, and the EG completed a questionnaire. Each week, the teacher lectured on two topics, and students used MS Word 2010 to practice in class.

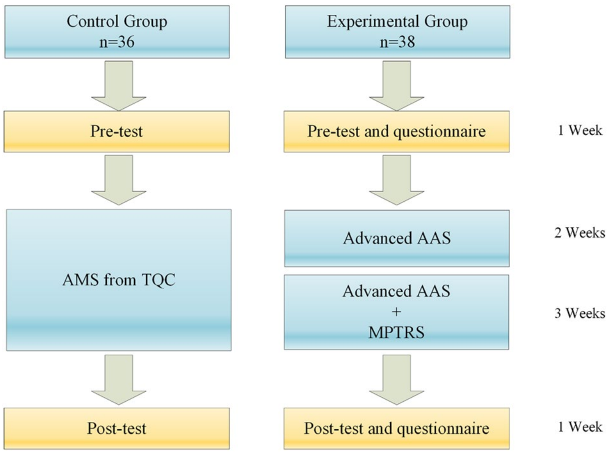

Fig. 5 Experimental schedule

Results and discussion

The EG used the advanced AAS and MPTRS in the practical and assessment phase, and the CG used the AMS in the limited version of TQC. SPSS (IBM 2012) was used to analyze data, and statistical analysis methods included independent-sample *t* test, paired-sample *t* test, one-way ANOVA, and regression analysis.

The CG students could ask for help from their classmates or teachers. The posttest scores of both groups were higher than the pretest scores. Independent-sample and paired-sample *t* tests were used to analyze the learning performance of students, as presented in Table 2. The independent-sample *t* test revealed no significant difference in pretest scores between the EG and CG ($t = 1.085$, $p = 0.28$). However, although the EG scored higher than the CG did in the posttest, the difference was nonsignificant ($t = -1.525$, $p = 0.13$). The reason may be that learning gaps between students in both the EG and CG were significant. The score differences within groups must be analyzed before and after the experiment. The paired-sample *t* test results in the EG reached significance ($t = 3.1$, $p = 0.04$), whereas those in the CG did not ($t = 0.665$, $p = 0.51$), indicating that our system can effectively improve students' learning performance. This is consistent with the results of Lee and Hu (2018), who demonstrated that peer learning can improve students' learning performance with grouping strategies. Students experiencing learning difficulties in the practical and assessment phase can receive help from their peers and enhance their learning.

The Pearson correlation was used to measure the strength of the linear relationship among posttest, the number of assignments uploaded in class, frequency of asking for help, and frequency of offering help to others.

Students uploaded their assignments to the server to receive marks and comments for correct and incorrect answers after completing their assignments. Therefore, they were allowed to upload their work several times to obtain their total scores. However, students with low learning performance were unable to complete their assignments and rarely uploaded files, and they easily lost motivation when encountering difficulties.

Table 3 indicates that posttest scores and the number of times that students uploaded assignments to the system were highly correlated. The teacher observed students who actively participated in the class and uploaded assignments several times to the system for marking could achieve higher scores. These results are similar to those of Murray (1998), who demonstrated that when students practice

Table 2 Learning performance comparison between control and experimental groups

	Experimental group ($n = 38$)	Control group ($n = 36$)	$t^b(p)$
Pretest	61.00 (±4.02)	67.00 (±2.89)	1.085 ($p = 0.28$)
Posttest	75.76 (±3.11)	68.89 (±3.27)	− 1.525 ($p = 0.13$)
$t^a(p)$	3.1 ($p = 0.04$)	0.665 ($p = 0.51$)	

[a]Paired-sample *t*-test

[b]Independent-sample *t* test

Table 3 Pearson correlation among variables after using the advanced automated assessment and machine learning-based peer tutor recommender systems

	Post-test	Number of assignments uploaded in class	Number of times asked for help (tutee)	Number of times offered help (tutor)
Post-test	1	0.358* $p=0.027$	− 0.076 $p=0.649$	0.307 $p=0.06$
Number of assignments uploaded in class	0.358* $p=0.027$	1	− 0.192 $p=0.249$	0.466** $p=0.003$
Number of times asked for help (tutee)	− 0.076 $p=0.649$	− 0.192 $p=0.249$	1	− 0.182 $p=0.274$
Number of times offered help (tutor)	0.307 $p=0.06$	0.466** $p=0.003$	− 0.182 $p=0.274$	1

**Correlation is significant at the 0.01 level (two-tailed)

*Correlation is significant at the 0.05 level (two-tailed)

more in class, they can achieve higher scores. Similarly, in our study, the students more participated in-class activities using our system, the more frequently they helped others. The frequency of uploading assignments is an essential factor for a teacher considering peer tutors, and our proposed model meets the same strategy.

Students were separated into four categories as follows, and the ANOVA comparison between tutors and tutees is presented in Table 4:

- 3: asks for help and helps others
- 2: is only a tutee
- 1: is only a tutor
- 0: never asks for help or helps others (never used the system).

Table 4 Analysis of variance comparison results for tutors and tutees based on four categories

Categories		Mean deference	Std. error	Sig
0	1	− 17.300	7.095	0.089
	2	13.282	6.922	0.241
	3	− 5.186	7.819	0.910
1	0	17.300	7.095	0.089
	2	30.582	6.932	0.001**
	3	12.114	7.819	0.420
2	0	− 13.282	6.932	0.241
	1	− 30.582	6.932	0.001**
	3	− 18.468	7.671	0.095
3	0	5.186	7.819	0.910
	1	− 12.114	7.819	0.420
	2	18.468	7.671	0.095

$p<0.05$, $**p<0.01$, $***p<0.001$

As shown in Table 4, the posttest scores of tutors in category 1 are significantly higher than those in category 2, indicating that students who had been only tutors recommended by the system scored significantly higher than those who were tutees only. During peer tutoring, students who act as tutors review and reinforce their learning (Bargh and Schul 1980; Whitman and Fife 1988). Because peer tutors must learn more, the tutors' performance is higher than that of students who do not teach. However, the scores of students in category 3 were not significantly higher than those of students in other groups may because when the system provides peer tutors, students select their familiars instead of high-performing students. This situation also matches the habits of students asking for help.

Recommender feedback analysis

To evaluate peer recommendation quality, each student receives a list of candidate tutors and completes tutor selection. The system then presents the user with a questionnaire. A description of the questionnaire points is presented subsequently. A conventional tutor assignment strategy considers only learning performance. However, social relationships are considered to include the peer tutor's computer application operating skills and relationship with the tutee in student feedback.

In the recommender feedback form, point 5 represents the best result, and point 1 represents the worst result. Our system allows a user to select a peer tutor candidate from the list of recommended results.

- 5 (excellent): This candidate has good computer application operating skills and is familiar with the tutee.
- 4 (good): This candidate has moderate computer application operating skills and is familiar with the tutee.
- 3 (good): This candidate has good computer application operating skills but is unfamiliar with the tutee.
- 2 (fair): This candidate has moderate computer application operating skills but is unfamiliar with the tutee.
- 1 (poor): None of the candidate tutors are suitable for the tutee.

The results were analyzed to reveal possible reasons for students to select tutors in our peer tutor system. In summary, 46% (31) of students selected peer tutors with good computer application operating skills who they were familiar with, and 46% (31) selected peer tutors with moderate computer application operating skills who they were familiar with. Totally, 92% (62) of peer tutors were familiar with their tutees. Only 3% (3) of tutees selected tutors were unfamiliar with them, and 2% (2) of tutees believed that the system could not provide a suitable candidate for them. This confirms that when students select a peer tutor, they prioritize friendship. None of the students selected a tutor who had good computer application operating skills who that was unfamiliar with, indicating that students prefer a peer tutor who they are familiar with and who has good computer application operating skills. This result is consistent with the findings of Ryan et al. (1997). Conventional methods of peer tutor grouping consider only learning performance and may group students who are unfamiliar with each other. However, our results demonstrated that students prefer to learn with their familiar classmates as peer tutors.

This study applied a machine learning algorithm to recommend tutors. In the first week, the records of students' social relationships and their preselected peer tutors were collected. During weeks 2 and 3, the students' learning performance was recorded. In week 4, students were allowed to start using our MPTRS to receive help from a peer tutor. The results are analyzed and summarized in Fig. 6. In week 4, peer tutors were recommended only nine times, and the recommended ranking result was M=4.33 and SD=0.82. In week 5, the recommended ranking result was M=4.27, SD=0.47 after 52 recommendations. This result was lower compared with that of week 3. It was assumed that the system had not collected enough input data to train the model. In week 6, the recommended ranking result has increased to M=4.71, SD=0.45) after 31 recommendations. Based on week 6 results, it was assumed that our model performed better with machine learning techniques.

The above quantitative analysis results show the machine learning technique is suitable to be used in the peer tutor recommendation approach. The results on the posttest showed that their computer application skills were significantly superior to those of students in the control group. The result revealed that the number of uploading assignments is significantly related to the number of helping others, shows that a higher activating, which is an important feature of peer tutors in the class. The peer tutors have higher learning performance than the tutees, and tutees' feedbacks shown the system recommending peer tutors matched their social relationships.

Questionnaire analysis

This study used the technology acceptance model (TAM) to analyze the perceived ease of use and usefulness for use and behavior intention (King and He 2006). Table 5 presents the results obtained from the questionnaire of students' attitudes toward automated assessment and peer learning in this system.

Questions Q1–Q3 were related to the automated assessment. According to the Q1 results (M=3.47, SD=1.10), students believed that the system interface and online instant scoring function are easy to use. The results of Q2 (M=3.58, SD=1.19)

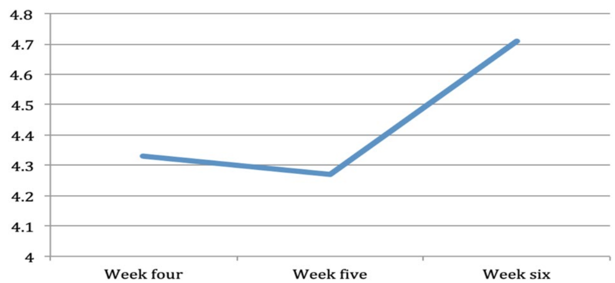

Fig. 6 Analysis of ranking results based on recommender feedback on the machine learning-based peer tutor recommender system

Table 5 Questionnaire developed to investigate ease of use and usefulness of the machine learning-based peer tutor recommender system

No	Questionnaire	Mean	SD
Q1	I think this system's interface is simple and easy to use	3.47	1.10
Q2	I think the web application is easy to use when uploading my work to the system	3.58	1.19
Q3	I think the online automated marking system is useful for learning	3.92	0.94
Q4	I think the system's tutor recommender module is easy to use	3.13	0.85
Q5	I am comfortable using the tutor recommender module	3.05	0.85
Q6	The tutor recommender module enhances my interaction with classmates	3.21	0.91
Q7	I feel that assigning a peer tutor helps me resolve difficulties in class	3.71	0.9
Q8	I feel that assigning a peer tutor helps me alleviate learning anxiety	3.61	1.12
Q9	I feel that assigning a peer tutor improves my attitude toward learning	3.75	1
Q10	I would like other courses to assign a peer tutor to help me	3.63	1.02
Q11	After teaching my classmates, I am more able to perform well in class	3.47	1.01

indicated that the system web application is easy to use when uploading files. Finally, the Q3 results (M = 3.92, SD = 0.94) validate the usefulness of the system in the learning process. The overall results indicated individual satisfaction with the advanced AAS due to several factors. For example, additional software applications are not required, and the web application interface is user-friendly for uploading assignments. As long as a screenshot is pasted in the web application, it can be automatically marked, and students can receive comments on incorrect parts.

The seven remaining questions (Q4–Q11) in Table 5 are related to peer tutoring. Students believed that the tutor recommender module is easy to use in our design (Q4, M = 3.13, SD = 0.85). The results of Q7 (M = 3.71, SD = 0.9) indicate that assigning a peer tutor can help students resolve difficulties. The tutor's computer application operating skills recommended by the system were higher than those of other students and can resolve tutees' difficulties. Students indicated that peer tutoring could reduce their learning anxiety (Q8, M = 3.61, SD = 1.12) and improve learning attitude (Q9, M = 3.75, SD = 1). Students agreed that peer tutoring increased interaction with classmates (Q6, M = 3.21, SD = 0.91), and they hoped that other courses could use the peer recommender system (Q10, M = 3.63, SD = 1.02). Moreover, students believed that they could improve their operational ability through teaching (Q11, M = 3.47, SD = 1.01). Students agree that using the recommender module felt comfortable but this item was lower than other items (Q5, M = 3.05, SD = 0.85). It is speculated that the possible reason is that the teacher asked the learning behind students to use this function.

Analysis of social relationships among students

To analyze social relationships among students in the EG, Gephi, an open-source social relationship analysis and visualization tool, was used (Bastian et al. 2009).

Figure 7 presents the results of our analysis. Text size indicates the number of times a student helped others (tutor), and node size indicates the number of

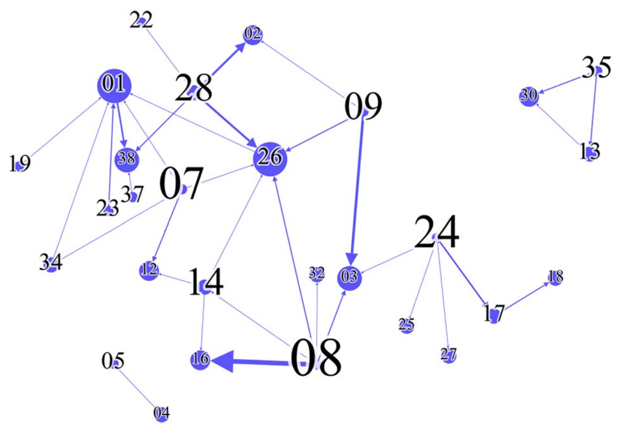

Fig. 7 Social relationships of students

times a student asked for help (tutee). Line width denotes the number of times peers interacted, and the direction of the arrow is from tutor to tutee. We also used an open questionnaire to select feedback from students who received help from a peer tutor. Students who were assisted were highly satisfied with their peer tutors and generally believed that peer tutors had the patience to assist students in their work. In Fig. 7, student 26 said that the peer tutors taught her patiently to help her complete the course quickly. Therefore, she provided five points on the feedback form. Student 03 said that she was busy at home with other coursework and had no time to practice word processing skills. However, the course provided an excellent opportunity to practice with classmates and sustain her progress. In addition, student 38 said that although she learns slowly, her peer tutor had taught her more. According to student 02, her classmates were kind and helped her finish her assignments on time.

Peer tutors can learn from teaching and can improve interpersonal relationships. However, the peer tutors proposed that tutees rely on peer tutor teaching instead of concentrating during the lectures.

Student 08 said, "I can identify my weaknesses, so I can learn more from teaching others." According to student 28, tutees should be asked to complete the assignment and use the marking system once. The peer tutor knew where the mistake occurred but was not required to teach them each step. According to student 24, peer tutoring helped develop relationships with classmates. In the other open-ended questionnaire, some students asked for help and helped others. Even though some students had low performance, they learned happily. Student 01 said, "I always feel that learning MS Word is quite difficult. The previous teachers' instruction was dull, so we did not learn much and felt no need to do any assignments. This class was completely different. Not only did the class atmosphere change, but the fun also increased a lot. Student 01 also said that he sometimes felt impatient when teaching, but after some time, he felt happy when teaching others, who will feel that he is a compassionate person.

Conclusion

Learning by doing strategy is widely used in different subject courses and some students experience challenging situations throughout their practice process. Peer learning is mutually beneficial when students share their knowledge, ideas, and experiences with other students. Automated marking is useful for students who can understand their incorrect answers and informs the teacher of student learning progress. Based on previous research, various peer tutoring strategies, and dynamic group formation systems for peer learning in class, in this study, an MPTRS with an advanced automated assessment approach was developed. The advanced AAS involved computer vision technology that flexibly evaluates student assignments and instantly returns feedback. Furthermore, the MPTRS was employed to manage mutual assistance among students by considering their social relationships, learning performance, and recommendation feedback to improve recommendations through machine learning techniques.

In the practical and assessment phase, students could review their assignments using the AAS. The teacher could observe student learning performance online. When students encountered difficulties, they could use the proposed system to request a peer tutor to help the student complete the assignment and solve the problem. In a conventional environment, the teacher cannot assist all students with learning difficulties in the practical and assessment phase. Our experimental method was divided into CG and EG. The CG used AMS in the limited version of a commercial AGS. In the practical and assessment phase the students used the AMS installed on each PC. If they encountered any difficulty in learning, they could seek assistance from peers or the teacher. The EG used our proposed system, which incorporates automatic marking and peer tutor recommendation with machine learning techniques. When a student requested assistance, the system recommended a peer tutor who can provide guidance.

The EG showed significantly improved learning performance between the pretest and posttest. Students had a positive attitude toward our automated assessment and believed that it is easy to use and useful for learning. Students indicated that assigning a peer tutor when students encounter difficulties can reduce learning anxiety and improve learning outcomes. Student feedback indicated that satisfaction with the recommended tutors was highest during the last week, demonstrating that our machine learning mechanism functioned well with the input of the students' social relationships, student performance, and recommendation feedback.

A higher correlation between frequency of helping others and student performance was observed. Presumably, the more these students help each other, the more their computer application operating skills improve. The proposed system recommended students who participate often as peer tutors. Moreover, the learning performance of students who only helped others was significantly higher than that of students who only asked for help from others. After analyzing student feedback on requesting help, tutor ranking results during the last week were the highest compared with those of previous weeks. The proposed system could identify an appropriate peer tutor for students who seek assistance.

Finally, the results demonstrated that students were satisfied both with the advanced AAS and the PRTS. The students who participated in this experiment majored in home economics, and the majority of them were girls. The EG and CG had only three and two male students, respectively. Therefore, gender differences could not be analyzed in this study. The teaching materials were based on MS Word and must be further verified for application in other computer application courses. The peer recommender in this experiment operated for only 3 weeks. Although students provided positive feedback, an extended evaluation is required.

Moreover, the tutor recommender system can be applied with learning by doing strategy to different courses such as mathematics, physical, and chemical, etc. Besides, computer vision also can use to automatically mark student work in computer application courses such as Excel, Power Points, accounting software, and other software applications.

References

Akobe, D., Popoola, S. I., Atayero, A. A., Oseni, O. F., & Misra, S. (2019). A web framework for online peer tutoring application in a smart campus. In *International conference on computational science and its applications, 2019* (pp. 316–326). New York: Springer.

Alegre, F., Moliner, L., Maroto, A., & Lorenzo-Valentin, G. (2019). Peer tutoring in algebra: A study in Middle school. *The Journal of Educational Research, 112*, 693–699.

Aronson, E., Wilson, T., Akert, R., & Sommers, S. (2016). *Social PSYCHOLOGY* (9th ed.). London: Pearson Education.

Bargh, J. A., & Schul, Y. (1980). On the cognitive benefits of teaching. *Journal of Educational Psychology, 72*(5), 593.

Bastian, M., Heymann, S., & Jacomy, M. Gephi: An open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media*.

Boud, D., Cohen, R., & Sampson, J. (2014). *Peer learning in higher education: Learning from and with each other*. New York: Routledge.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. Sebastopol, CA: O'Reilly Media Inc.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331–370.

Chao, I.-M. (2007). *The study on the information abilities and e-learning opportunities for junior high school students at Taipei City (Unpublished master's thesis)*. Taipei City: National Normal University.

Chorana, A., Lakhdari, A., Cherroun, H., & Oulad-Naoui, S. (2015). XML-based e-assessment system for Office skills in open learning environments. *Research and Practice in Technology Enhanced Learning, 10*(1), 12.

Chu, H.-C., Chen, J.-M., Hwang, G.-J., & Chen, T.-W. (2019). Effects of formative assessment in an augmented reality approach to conducting ubiquitous learning activities for architecture courses. *Universal Access in the Information Society, 18*, 221–230.

Chuang, C.-W., Hwang, G.-J., & Tsai, W.-J. (2018). A peer tutoring-based concept mapping approach to improving students' learning achievements and attitudes for a social studies course. *International Journal of Online Pedagogy and Course Design (IJOPCD), 8*(1), 1–12.

Computer Skill Foundation (2019). www.csf.org.tw.

Conrad, E. (1974). Peer tutoring: A cooperative learning experience.

Creswell, J. W. (2008). *Educational research: Planning, conducting, and evaluating quantitative*. Upper Saddle River, NJ: Prentice Hall.

Damon, W., & Phelps, E. (1989). Critical distinctions among three approaches to peer education. *International Journal of Educational Research, 13*(1), 9–19.

Day, M.-Y., & Lin, Y.-D. (2017). Deep learning for sentiment analysis on google play consumer review. In *2017 IEEE international conference on information reuse and integration (IRI), 2017* (pp. 382–388). IEEE.

Diaby, M., & Viennet, E. (2014). Taxonomy-based job recommender systems on Facebook and LinkedIn profiles. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), 2014* (pp. 1–6). IEEE.

Diaby, M., Viennet, E., & Launay, T. (2013). Toward the next generation of recruitment tools: An online social network-based job recommender system. In *2013 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2013), 2013* (pp. 821–828). IEEE.

Dong, J.-J., & Hwang, W.-Y. (2012). Study to minimize learning progress differences in software learning class using PLITAZ system. *Educational Technology Research and Development, 60*(3), 501–527.

Dong, J.-J., Hwang, W.-Y., Shadiev, R., & Chen, G.-Y. (2018). Implementing on-call-tutor system for facilitating peer-help activities. *IEEE Transactions on Learning Technologies, 12*(1), 73–86.

Duran, D. (2017). Learning-by-teaching. Evidence and implications as a pedagogical mechanism. *Innovations in Education and Teaching International, 54*(5), 476–484.

Evans, M. J., & Moore, J. S. (2013). Peer tutoring with the aid of the Internet. *British Journal of Educational Technology, 44*(1), 144–155.

Falchikov, N. (2001). *Learning together: Peer tutoring in higher education*. London: Psychology Press.

Fantuzzo, J. W., Davis, G. Y., & Ginsburg, M. D. (1995). Effects of parent involvement in isolation or in combination with peer tutoring on student self-concept and mathematics achievement. *Journal of Educational Psychology, 87*(2), 272.

Gibbs, S. F., Steel, G. D., & McKinnon, A. (2015). *Are workplace end-user computing skills at a desirable level? A New Zealand perspective*.

Good, T. L., & Lavigne, A. L. (2017). *Looking in classrooms*. New York: Routledge.

Goodlad, S., & Hirst, B. (1989). *Peer tutoring. A guide to learning by teaching*. Williston Park: ERIC.

Greenwood, C. R., Arreaga-Mayer, C., Utley, C. A., Gavin, K. M., & Terry, B. J. (2001). Classwide peer tutoring learning management system: Applications with elementary-level English language learners. *Remedial and Special Education, 22*(1), 34–47.

Greenwood, C. R., & Delquadri, J. (1995). Classwide peer tutoring and the prevention of school failure. *Preventing School Failure: Alternative Education for Children and Youth, 39*(4), 21–25.

Greenwood, C. R., Delquadri, J. C., & Hall, R. V. (1989). Longitudinal effects of classwide peer tutoring. *Journal of Educational Psychology, 81*(3), 371.

Gulli, A., & Pal, S. (2017). *Deep learning with keras*. Birmingham: Packt Publishing Ltd.

Harger, R. O. (1996). Teaching in a computer classroom with a hyperlinked, interactive book. *IEEE Transactions on Education, 39*(3), 327–335.

Hawkins, J., Crim, C., Ganz, J. B., & Kennedy, K. D. (2019). The power of peers: Setting a course for 21st century skills in inclusive classrooms. In *Cultivating inclusive practices in contemporary K-12 education* (pp. 66–89). IGI Global.

Hill, T. G. (2011). Word grader and powerpoint grader. *ACM Inroads, 2*(2), 34–36.

IBM. (2012). *SPSS version 21.0*. Armonk, NY: IBM Corp.

Ketele, P., Jacobs, A., Boruett, N., & Derese, A. (2010). PerSIST: A PAL system for clinical skills training: A planning and implementation framework: Guide supplement 30.8: Practical application. *Medical Teacher, 32*(9), 782–784.

King, W. R., & He, J. (2006). A meta-analysis of the technology acceptance model. *Information & Management, 43*(6), 740–755.

Klašnja-Milićević, A., Vesin, B., Ivanović, M., & Budimac, Z. (2011). e-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education, 56*(3), 885–899.

Koike, H., Akama, K., Ishikawa, C., Nakamura, N., & Watanabe, S. (2007). Toward a software development model for automatic marking software. In *Proceedings of the 35th annual ACM SIGUCCS fall conference, 2007* (pp. 190–193). ACM.

Lacaba, A. B., Magalona, J. D., & Lacaba, T. V. G. (2018). Peer teaching strategy to increase mathematics proficiency level (an action research) (Eastern Samar State University). *International Journal of Scientific Research and Management, 6*(08), 2079–2085.

Lahtonen, T., & Isomöttönen, V. (2012). PARSI: a tool for automatic assessment of office documents and basic IT skills. In *Proceedings of the 12th Koli calling international conference on computing education research, 2012* (pp. 174–180). ACM.

Lee, C.-I., & Hu, C.-W. (2018). Assisting students with learning difficulties in class by using dynamic grouping strategy with peer learning. *Journal of Liberal Arts and Social Sciences, 14*(2), 117–137.
Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing, 1*, 76–80.
Long, S., Dowsing, R., & Craven, P. (2003). Knowledge-based systems for marking professional IT skills examinations. In *Applications and Innovations in intelligent systems X* (pp. 3–16). New York: Springer.
Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems, 74*, 12–32.
Madaio, M., Peng, K., Ogan, A., & Cassell, J. (2018). *A climate of support: A process-oriented analysis of the impact of rapport on peer tutoring*. Brunswick: International Society of the Learning Sciences Inc.
Matthews, K., Janicki, T., He, L., & Patterson, L. (2012). Implementation of an automated grading system with an adaptive learning component to affect student feedback and response time. *Journal of Information Systems Education, 23*(1), 71.
McKellar, N. A. (1986). Behaviors used in peer tutoring. *The Journal of Experimental Education, 54*(3), 163–167.
Mihaescu, M. C., Popescu, P.-S., & Ionascu, C. M. (2015). *Intelligent tutor recommender system for on-line educational environments* (pp. 516–519). Paris: EDM.
Murray, T. (1998). Authoring knowledge-based tutors: Tools for content, instructional strategy, student model, and interface design. *The Journal of the Learning Sciences, 7*(1), 5–64.
Nassar, N., Jafar, A., & Rahhal, Y. (2020). A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems, 187*, 104811.
Ochirbat, A., Shih, T. K., Chootong, C., Sommool, W., Gunarathne, W., Wang, H.-H., et al. (2018). Hybrid occupation recommendation for adolescents on interest, profile, and behavior. *Telematics and Informatics, 35*(3), 534–550.
OpenCV library (2019). http://opencv.org.
Puri, R., & Jain, V. (2019). Barcode detection using OpenCV-python. *Science, 4*(1), 97–99.
Rahutomo, R., Perbangsa, A. S., Soeparno, H., & Pardamean, B. (2019). Embedding model design for producing book recommendation. In *2019 International conference on information management and technology (ICIMTech), 2019* (Vol. 1, pp. 537–541). IEEE.
Ren, X., Tang, F., & Tang, P. (2010). Word operation automatic marking system based on VBA. *Computer Knowledge and Technology, 21*, 059.
Riad, A., Sporer, C., Bukhari, S. S., & Dengel, A. (2017). Classification and information extraction for complex and nested tabular structures in images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR), 2017* (Vol. 1, pp. 1156–1161). IEEE.
Rohrbeck, C. A., Ginsburg-Block, M. D., Fantuzzo, J. W., & Miller, T. R. (2003). Peer-assisted learning interventions with elementary school students: A meta-analytic review. *Journal of Educational Psychology, 95*(2), 240.
Ryan, A. M., Hicks, L., & Midgley, C. (1997). Social goals, academic goals, and avoiding seeking help in the classroom. *The Journal of Early Adolescence, 17*(2), 152–171.
Sarkar, D., Bali, R., & Ghosh, T. (2018). *Hands-on transfer learning with python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Birmingham: Packt Publishing Ltd.
Shepitsen, A., Gemmell, J., Mobasher, B., & Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on recommender systems, 2008* (pp. 259–266). ACM.
Sinha, A., Bayer, J., & Bukhari, S. S. (2019). Table localization and field value extraction in piping and instrumentation diagram images. In *2019 international conference on document analysis and recognition workshops (ICDARW), 2019* (Vol. 1, pp. 26–31). IEEE.
Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks, 2*(6), 568–576.
Swithenby, S. J. (2006). E-assessment for open learning. In *European Association of Distance Teaching Universities (EADTU) Annual Conference, 2006*.
Topping, K. (2000). *Tutoring*. Geneva: International Academy of Education Genf.
Topping, K., & Ehly, S. (1998). *Peer-assisted learning*. New York: Routledge.

Wang, X., Zhang, Y., Yu, S., Liu, X., Yuan, Y., & Wang, F.-Y. (2017). E-learning recommendation framework based on deep learning. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 455–460). IEEE.

Whitelock, D. M., & Brasher, A. (2006). *Developing a roadmap for e-assessment: Which way now?*. Loughborough: Loughborough University.

Whitman, N. A., & Fife, J. D. (1988). *Peer teaching: To teach is to learn twice. ASHE-ERIC higher education report no. 4*. Williston Park: ERIC.

Wongwatkit, C., Srisawasdi, N., Hwang, G.-J., & Panjaburee, P. (2017). Influence of an integrated learning diagnosis and formative assessment-based personalized web learning approach on students learning performances and perceptions. *Interactive Learning Environments, 25*(7), 889–903.

Xie, H., Chu, H.-C., Hwang, G.-J., & Wang, C.-C. (2019). Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017. *Computers & Education, 140*, 103599.

Zaíane, O. R. (2002). Building a recommender agent for e-learning systems. In *International Conference on Computers in Education, 2002. Proceedings* (pp. 55–59). IEEE.

Zhang, L., Luo, T., Zhang, F., & Wu, Y. (2018). A recommendation model based on deep neural network. *IEEE Access, 6*, 9454–9463.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2017). Deep learning based recommender system: A survey and new perspectives (2017). http://arxiv.org/abs/1707.07435.

Zhang, Y., Yang, C., & Niu, Z. (2014). A research of job recommendation system based on collaborative filtering. In *2014 Seventh International Symposium on Computational Intelligence and Design, 2014* (Vol. 1, pp. 533–538). IEEE.

Zhu, Y., & Shen, F. (2013). An office automatic marking system research and implementation. *Journal of Theoretical & Applied Information Technology, 47*(1), 242–245.

Zhao-Heng Ma is a computer teacher of Taoyuan Municipal Zhongli Home Economics and Commercial Senior High School and also a Ph.D. student of Department of Computer Science & Information Engineering, National Central University, Taiwan.

Wu-Yuin Hwang is a professor and the director of Graduate Institute of Network Learning Technology, National Central University, Taiwan.

Timothy K. Shih is a professor and the vice dean of the college of EECS at the National Central University, Taiwan.