AI-Based Diagnostic Assessment System: Integrated With Knowledge Map in MOOCs

Chia-An Lee, Nen-Fu Huang, *Senior Member, IEEE*, Jian-Wei Tzeng, and Pin-Han Tsai

***Abstract*—Massive open online courses offer a valuable platform for efficient and flexible learning. They can improve teaching and learning effectiveness by enabling the evaluation of learning behaviors and the collection of feedback from students. The knowledge map approach constitutes a suitable tool for evaluating and presenting students' learning performance levels. This study proposes an artificial-intelligence-based knowledge assessment system that integrates knowledge maps to determine students' familiarity with and mastery of course contents. This study employs a structural approach encompassing data collection, data preprocessing, model training, testing, and evaluation. In detail, the system can then customize the knowledge maps and recommend videos according to the knowledge nodes. Students consequently dedicate additional time to studying concepts with which they are unfamiliar and adjust their learning efforts accordingly. After teachers and teaching assistants have captured students' performance metrics and idiosyncratic weaknesses through knowledge maps, teachers can modify the teaching materials. Through the use of education data mining and learning analytics, our system can benefit both teachers and online learners. We hope that the proposed system provides a more personalized and intelligent online learning environment within which students can learn in a more efficient and flexible manner.**

***Index Terms*—Artificial intelligence (AI), deep learning, knowledge map, learning analytics, massive open online courses (MOOCs).**

I. Introduction

MASSIVE open online courses (MOOCs) are open-access educational resources that are available in every field for online learners worldwide. MOOCs not only feature high-quality instructional videos created by professors and lecturers from prestigious universities but also have numerous learning modes, including live video streaming lectures, efficient assessments, and discussion forums. Existing MOOC platforms include Coursera, edX, Udacity, and Khan Academy.

Manuscript received 25 March 2020; revised 4 June 2023 and 14 August 2023; accepted 17 August 2023. Date of publication 24 August 2023; date of current version 16 October 2023. This work was supported in part by the Ministry of Education of Taiwan under Grant PEE1120553, in part by the National Science and Technology Council of Taiwan under Grant NSTC 108-2221-E-007-062-MY3, Grant NSTC111-2410-H-025-039, and Grant NSTC 112-2410-H-025-019, and in part by National Tsing Hua University, Taiwan. *(Corresponding author: Jian-Wei Tzeng.)*

Chia-An Lee and Nen-Fu Huang are with the Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan (e-mail: ck_star2001@hotmail.com; nfhuang@cs.nthu.edu.tw).

Pin-Han Tsai is with the Communication Engineering Institute, National Tsing Hua University, Hsinchu 300, Taiwan (e-mail: k52276hank@gmail.com).

Jian-Wei Tzeng is with the Department of Information Management, National Taichung University of Science and Technology, Taichung 404, Taiwan (e-mail: tjw@nutc.edu.tw).

Digital Object Identifier 10.1109/TLT.2023.3308338

MOOCs are considered a modern method of education because they provide a self-directed learning environment and are convenient for learners. Because of the increasing number of Internet users enrolling in MOOCs, a substantial number of online users' learning records and patterns can be collected and analyzed. Big data and artificial intelligence (AI) technology have become prominent research topics in various fields, including machine learning and data science. Cases of AI applications in learning have increased recently. For example, early researchers have focused on creating personalized teaching systems for solitary learners; by contrast, recent researchers have considered multiple learners together in a single learning context [1]. Some AI and cognitive science techniques can improve practitioners' understanding of the nature of learning and teaching and enable them to build systems to help learners gain new skills or understand new concepts [2], [3]. Consequently, AI has begun to affect student experiences because it enables the collection of data on student learning and the analysis of that data to provide feedback for course designs.

Although the self-regulated learning structures of MOOCs offer considerable flexibility and resources of interest, numerous learners fail to complete courses because of the pressure-free learning environment. Additionally, remarkable disorientation is observed when an individual attempts to acquire knowledge; this can be attributed to the substantial amount of learning material available to users [4], [5], [6], [7]. Learners also face difficulties if they lack sufficient prior knowledge to understand the course material [8].

In recent years, knowledge maps have been extensively utilized in the fields of education and business. A knowledge map is a visual representation of available knowledge and the pattern of knowledge flow [9]. Such knowledge maps can be used to analyze and summarize information efficiently and thereby enhance the learning experience. Evidence suggests that creating or referring to summarized data enhances the ability to recall the summarized information [10], [11]. Additionally, knowledge maps are more effective than text passages in assisting learners to recollect central ideas and details [12].

In MOOCs, the most common method for evaluating students' learning progress is the administration of exams with multiple-choice or open-ended questions. Relevant studies [13], [14], [15] have developed methods for progress evaluation by assessing the knowledge maps generated by students. Several studies have proposed algorithms for predicting student performance with behavior-based machine learning features [16], [17], [18]. However, most MOOC students participating in video-watching seldom complete the quizzes or exams, which considerably limits the effectiveness of the performance evaluation. Instructors face considerable difficulty in evaluating students' achievement precisely. To address these problems, we propose a novel semiautomatic tool called a semiautomatic knowledge map generator (SEAKMAP). Concurrently, we have developed an AI assessment system that leverages deep neural network (DNN) architecture. This choice was made to ensure the reliability and transparency of the model while still capitalizing on the benefits of neural networks in analyzing students' video-watching behavior.

Our objective was to create an AI-supported online learning environment to improve learning. Personalized service [19], which considers the abilities, preferences, and browsing behaviors of individual students, is highly crucial in MOOCs and provides individual learning paths for students. We developed data-driven systems to promote self-paced learning by using educational data mining. All systems were tested and integrated into a live MOOC platform, namely ShareCourse, in Taiwan. The results of the tests were highly positive and can serve as a reference for related systems.

II. System Architecture

This section introduces the entire system architecture of SEAKMAP, the item difficulty and discrimination system, and the AI assessment system. Fig. 1 illustrates the architecture of the AI assessment system, which combines an item difficulty and discrimination system with a knowledge map system.

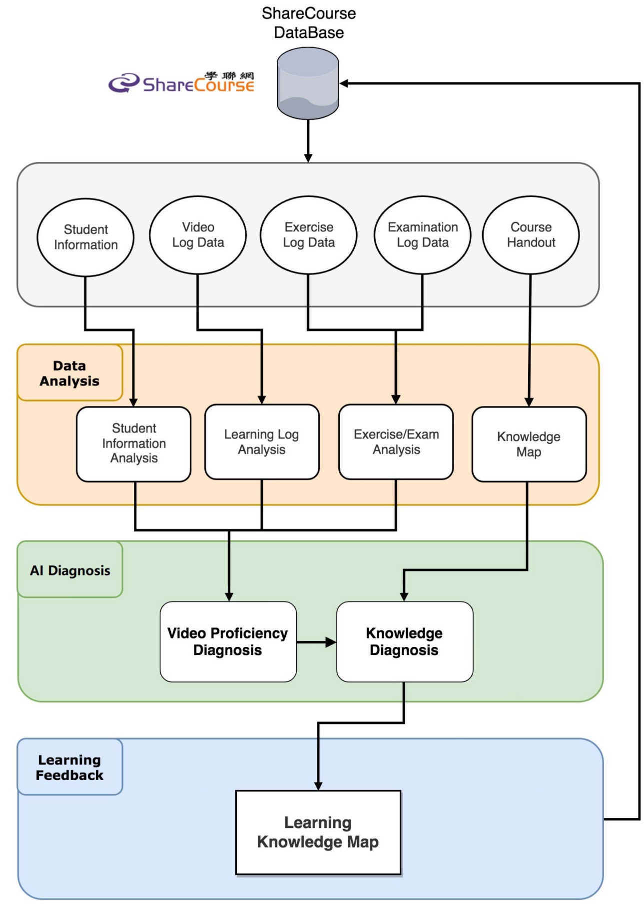

Fig. 1. System architecture.

A. ShareCourse

ShareCourse, one of the largest MOOC platforms in Taiwan, offers more than 1000 courses and has had more than 100 000 user registrations since 2012. ShareCourse collaborates with more than 70 universities and colleges in Taiwan and provides not only an efficient teaching and learning environment for online lecturers and learners but also valuable supplementary resources such as online discussion rooms, self-learning performance visualization, and knowledge maps. Some courses also provide certificates and recommendation letters, which can advance students' careers.

B. SEAKMAP

We propose SEAKMAP to optimize human labor and the time required for preparing knowledge maps. SEAKMAP is used to semiautomatically construct knowledge maps by analyzing course handout files, which are considered to be conceptual models for teachers.

Knowledge maps are node-link representations where ideas from a given knowledge domain are represented as nodes, and the nodes are connected through a series of links [20]. Because of the enrichment of information resources and excessive complexity of relationships between knowledge in recent years, knowledge maps that could clearly express knowledge nodes and knowledge links have become a crucial research topic [21], [22]. Numerous studies have demonstrated that knowledge maps have positive effects on learners' performance levels.

Knowledge maps, similar to other representations such as concept maps and graphic organizers, visualize the relationships between knowledge, and thereby provide an intuitive way to understand such relationships [23]. However, knowledge maps can have a specific series of links that connect nodes [24]. For example, because the links of a knowledge map are based on relevancy, and thus, a link between nodes can be an elaborative link that extends information, rather than just concepts. Thus, our system provides users with examples of a concept by allowing them to click on the nodes, and thereby find related videos and materials.

SEAKMAP comprises two parts: a user interface (UI) and a handout analysis module. The UI enables users to upload their course handouts, select keywords and titles, compose drafts, and finalize modifications and adjustments to obtain perfected knowledge maps.

The handout analysis module performs two operations: 1) keyword extraction and 2) relation extraction. Keyword extraction involves extracting keywords from course handout files by using a term-frequency–inverse document frequency (TF–IDF) algorithm [25]. Relation extraction entails analyzing the course handout files to obtain hidden information and defining

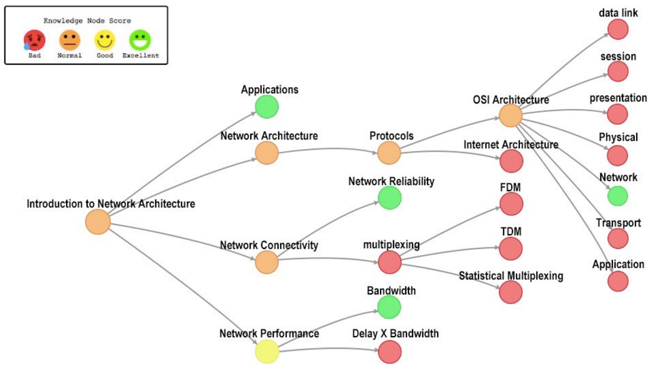

Fig. 2. Example of colored knowledge map.

the relationships between the keywords extracted during the previous processes.

After the final knowledge map has been generated, the system must map the concepts of each knowledge node to the relevant course material (e.g., course videos, course exercises, course quizzes, and course questions from exams). Material mapping must rely on the help of experts in the course, such as course instructors or teaching assistants (TAs).

C. Item Analysis System

ShareCourse has an item analysis system to identify alternative concepts from students' responses to exercises in accordance with item response theory (IRT). In IRT, difficulty and discrimination parameters are calculated on the basis of students' answer logs for exercises.

A high difficulty number means that an exercise is relatively easy, and a low difficulty number means that an exercise is relatively difficult. A high discrimination number indicates that students in a high-score group should answer questions correctly (and vice versa).

D. AI Assessment System

The AI assessment system utilizes learning logs from students to calculate the knowledge score of each knowledge node. These scores are further integrated with knowledge maps and displayed on the UI in four different colors (see Fig. 2). Each color represents the level of a set of knowledge scores that students received. Students can monitor their learning progress easily through this evaluation mechanism.

III. System Implementation

This section introduces the implementation algorithms of the proposed systems, namely the AI assessment system, SEAKMAP system, and item difficulty and discrimination system; this section also details the UI design.

This study employs a structural approach encompassing data collection, data preprocessing, model training, testing, and evaluation.

This approach begins with data collection, where the researchers gather relevant information from students' activities on the ShareCourse platform, specifically their viewing and answering behaviors.

Next, the collected data undergoes a series of preprocessing steps to ensure its quality and suitability for analysis, including data cleanup that addresses inconsistencies or errors in the data; data normalization that brings the data into a standardized format, facilitating further analysis; and feature extraction, identifying, and extracting meaningful features from the collected data. To enhance the feature extraction process, SEAKMAP and an item analysis system are utilized, allowing for the extraction of additional relevant features.

Third, we use DNNs for model training and testing. These networks are trained on the collected and preprocessed data to learn patterns and relationships within the dataset. During the testing phase, we adopt a technique known as 10-fold cross-validation. It involves splitting the data into 10 subsets, training the model on nine subsets, and evaluating its performance on the remaining subset. This process is repeated 10 times to ensure comprehensive testing and robust evaluation.

Finally, the performance of the trained model is evaluated by the confusion matrix and its derivatives. The confusion matrix provides insights into the model's accuracy and its ability to correctly classify instances. By analyzing the confusion matrix, researchers can assess the performance of the model and identify areas that may require improvement.

A. SEAKMAP

The operating procedure of SEAKMAP is divided into three critical steps: 1) keyword extraction, 2) relation extraction, and 3) knowledge map generation.

The keyword extraction process is conducted to extract keywords from the course handouts. First, text segmentation is executed to divide the text into meaningful fragments. Direct word division based on space is suitable for English sentences. However, for Chinese sentences, Jieba [26] open-source tools must be used to obtain meaningful text after segmentation. Second, irrelevant words are removed from the segmentation results based on the stop-word list [27]. We adjust the stop-word list to make it suitable for Chinese stop-words.

In the second step, we use an *n*-gram method to divide the English words and Chinese nouns separated by the word segmentation process in the previous step. We set *n* to 1–5 to find all occurrences of word combinations, meaning we seek terms with lengths ranging from one to five. An *n*-gram algorithm is based on a statistical language model [28]. An *n*-gram is a set of *n*-item subsequences from a given sequence, usually used for counting the frequency of words appearing together in a fixed order. For example, a 2-gram for "I like food." would be "I like" and "like food." However, in our proposed system, we use the *n*-gram method to calculate only the number of occurrences rather than the frequency of the word combinations. Once the calculation is complete, we obtain a list of the number of word occurrences.

Next, we use the word occurrence numbers from this list to calculate the TF–IDF value of each word or term in the list. The purpose of calculating the TF–IDF values is to find the

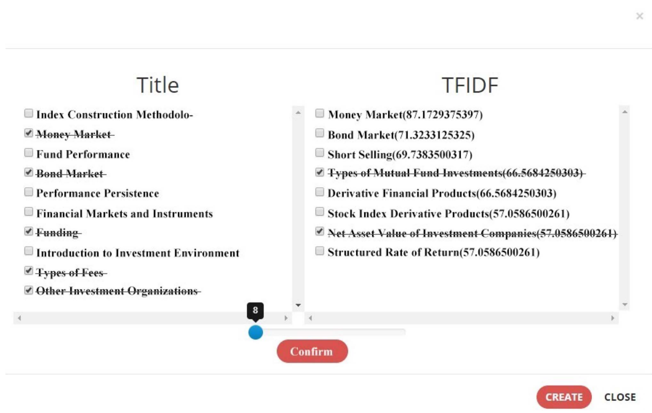

Fig. 3. Result obtained using the TF-IDF method, displayed on the UI.

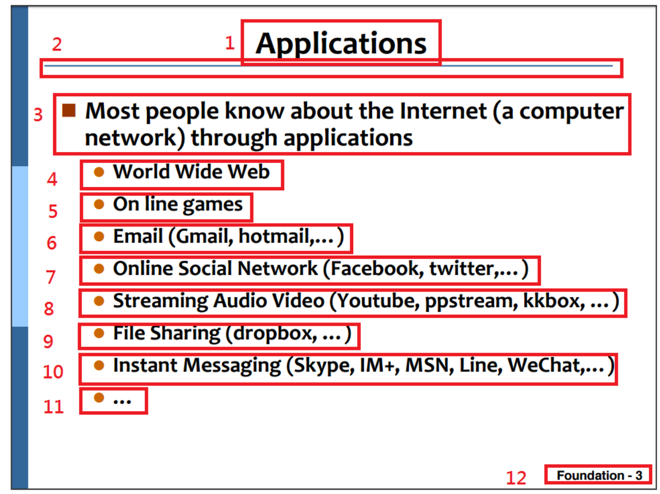

Fig. 4. Handouts analysis example.

keywords in the handouts; this is because the TF–IDF weighting of a word or a term indicates its overall importance in the handout files. The higher the TF–IDF value of a word, the greater the importance of its meaning in all handouts and the more likely it is to be a keyword. The following equation presents the formula for TF–IDF calculation:

$$\mathrm{tfidf}(t, d, D) = \frac{f_{t,d}}{\sum_k f_{k,d}} \cdot \log \frac{N}{1 + |\{d \in D : t \in d\}|} \tag{1}$$

where $f_{t,d}$ is the number of times a particular word appears in a handout, $\sum_k f_{k,d}$ is the number of times all words appear in a handout, N is the total number of handouts, and $|\{d \in D : t \in d\}|$ is the number of handouts containing the particular word.

After calculating the TF–IDF value, we examine the words and terms in more detail. First, we find the words or terms with the same TF–IDF value. If the matching words are covered or duplicated, we use the longer one. For example, as "TCP" appears in "TCP IP" and both have the same TF–IDF, we replace "TCP" with "TCP IP" to reduce the number of redundant words.

The weighting result and the titles of the weekly handouts are both displayed on the UI (see Fig. 3). On this UI, the TA and the teacher can select the desired number of keywords by adjusting a scroll bar. Then, they can tick the checkbox to delete any titles and keywords that are not required to be nodes in the knowledge map. The remaining titles and keywords are retained for the next step: analyzing the course handouts to obtain the relationships between titles and keywords.

Subsequently, the relation must be extracted, which entails analyzing the course handouts to obtain hidden information, such as text location and font size, to determine the relationships between the chosen keywords or sentences. The text layer is identified on each page of the course handouts on the basis of text locations and font size. Based on the hierarchy, the relationship between the top-to-bottom layers of the text on one page is obtained. As illustrated in Fig. 4, the text "Applications" (object 1) is on the top layer because its location is at the top and its font size is the largest. After the top layer is determined, the other layers must be determined from the distance between the text and the left page edge. Object 2 is ignored because it is not a text object. Object 3 is the second layer because it is the first text object after the top layer. Object 4 is the third layer because the distance between it and the left edge is greater than that of object 3. Objects 5–11 are also regarded as the third layer because they are the same distance from the left edge, which is greater than that of object 3. Thus, every text object's layer is determined on each page of the handouts.

We designed an algorithm to determine the relationships between different objects. Fig. 5 illustrates an example of relation extraction. On page 1, object A is in the top layer, objects B and D are in the second layer, and objects C and E are in the third layer. On page 2, no top layer is present because no title is present on the page; object F is in the second layer, and object G is in the third layer.

On the basis of the acquired hierarchy, we iterate each object with respect to the layer to identify the relationships between all objects. Object A is recognized as the root because it stands alone in the top layer. Object B is in the second layer, and the program checks the previous and following layers before finalizing the parent—child relationship. The algorithm recognizes the relationship between objects B and A, which is the root of the previous layer, and with object C, which is in the next layer. Thus, we obtain the relationships A–B and B–C. When the iteration reaches C, the machine identifies that object C follows the previous object B in the second layer and that no object is present in the next layer. Therefore, no further child relationships exist for C. The relationships between the remaining objects are determined in a similar manner. The process ceases when the algorithm finds another title. In summary, the algorithm defines the hierarchy and connections between all objects (see Fig. 6).

1) UI Platform: On the UI platform, users can go through the entire process of generating a knowledge map, such as uploading course handouts, selecting keywords, and editing the knowledge map results of an article. Fig. 7 illustrates the editing page on which users can view and edit the knowledge map results of their manuscripts.

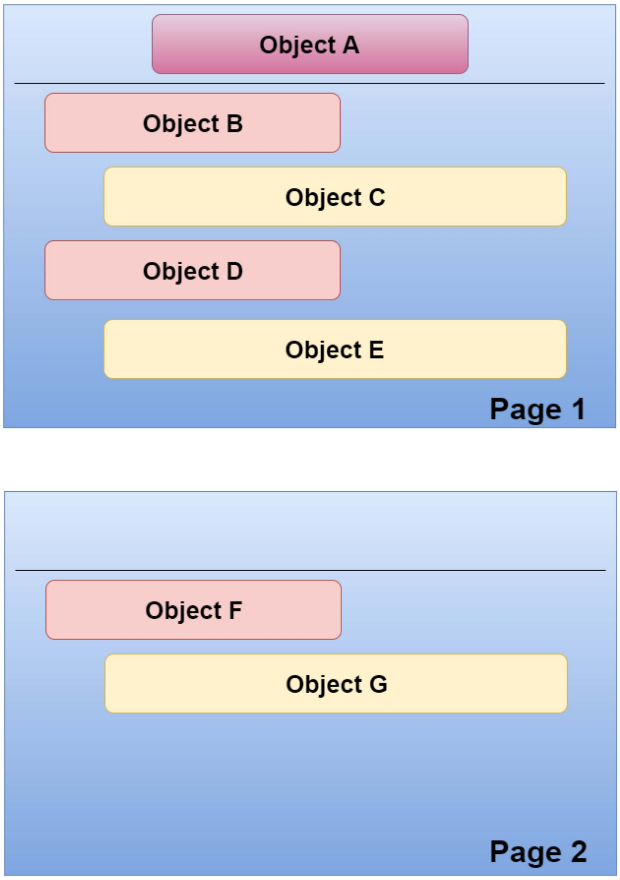

Fig. 5. Example of relation extraction.

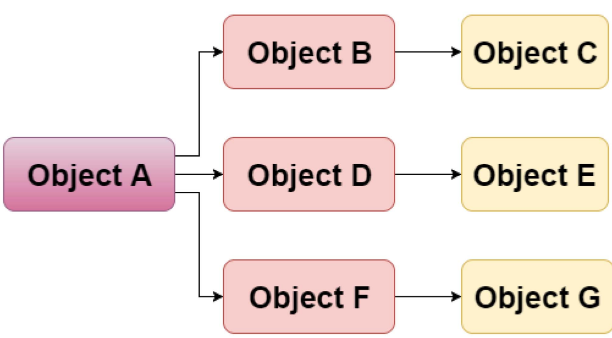

Fig. 6. Relationships extracted for example in Fig. 5.

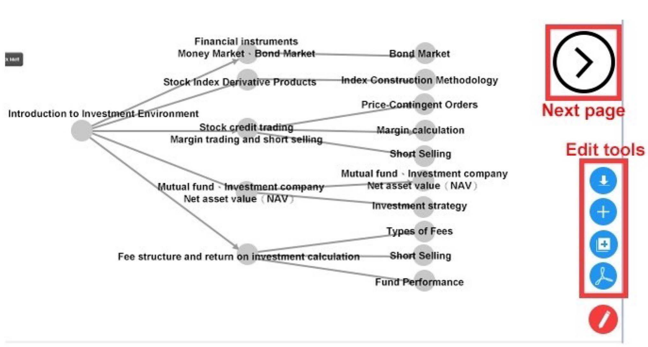

Fig. 7. Knowledge map editing page.

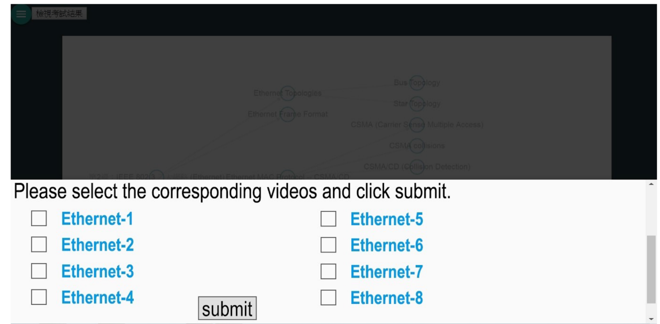

Fig. 8. Editing video link page.

Moreover, editing tools are located in the bottom-right corner, enabling users to add and delete nodes, create relationships between nodes, and download a portable network graphics file of the knowledge map. After the knowledge map generation is completed, TAs and instructors can click the "Next page" button on the web page to insert video hyperlinks into each node (see Fig. 8). This enables students to click the nodes on the knowledge map and access related videos.

B. Item Analysis System

This section describes the algorithm that analyzes students' answer records and derives the item difficulty index and item discrimination index, which are implemented on an existing MOOC platform.

IRT, representing a statistical model for evaluating the quality of measures [29], improves assessment quality, increases the efficiency of the testing process, and provides comprehensive testing guidelines [30]. Item analysis is proposed to enhance teaching and classify students according to their test performance so that teachers can focus on improving this performance [31]. Item difficulty and item discrimination are also considered to be practical methods for estimating item quality.

Our item analysis system employs Kiat's [31] definitions for the difficulty index and discrimination index. To calculate the difficulty index and discrimination index, we can consider data from 27% of high-achieving students and 27% of low-achieving students for the item analysis. Wiersma and Jurs [32] recommended using 27% because this value maximizes the differences in normal distributions and provides a sufficient number of cases for analysis. If a student is in the top 27% in terms of score, then the student is categorized as belonging to the "high" group, whereas students in the bottom 27% are placed in the "low" group. The analyzer aggregates the records for each problem and group according to these labels. Finally, the following equations are employed to determine the item difficulty and item discrimination indices:

$$\text{difficulty index} = \frac{P_H + P_L}{2} \tag{2}$$

$$\text{discrimination index} = P_H - P_L \tag{3}$$

where P_H and P_L are the proportions of students who answered the item correctly in the high and low groups, respectively.

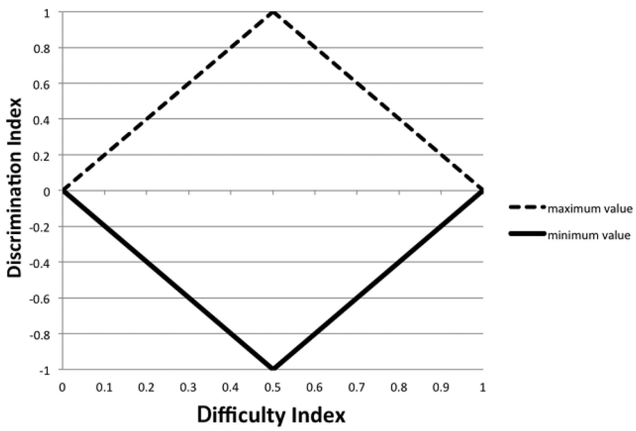

Fig. 9. Relationship between discrimination and difficulty indices.

The difficulty index is within the range [0, 1]; a high difficulty index indicates that the item is relatively easy. The discrimination index is within the range [−1, 1]; a high discrimination index indicates a relatively high ability to distinguish between students in the high-score and low-score groups. Specifically, students in the high-score group should answer the problem correctly if the problem has a high discrimination index. In general, a difficulty index near 0.5 is considered appropriate. Furthermore, the discrimination index is bounded by the difficulty index and reaches the maximum possible value when the difficulty index is 0.5 (see Fig. 9).

C. AI Assessment System

In this section, we explain how to use students' learning records and knowledge maps to implement and predict the score of each knowledge node. Similar approaches have been published. Yang et al. [33] proposed a method for using time-series neural networks to personalize prediction for the first correct attempt at a challenge. Moreover, Li et al. [34] proposed a model for predicting test scores; they selected 15 features in accordance with several educational theories and used all of the selected features to predict quiz grades.

The primary idea behind our system is that for a particular knowledge concept, if a student can answer a relatively difficult question after performing a series of learning behaviors, then he or she is considered to be familiar with the concept. We use deep learning to predict whether students will correctly answer questions at certain difficulty levels and degrees of discrimination.

1) Feature Extraction: Videos are often the primary learning method in MOOCs. However, even if a user watches a video, it does not mean that he or she fully understands the concept presented therein. Therefore, our system uses the user's learning log and deep learning to ascertain user proficiency for various concepts.

We collect data concerning each user's actions, time required for watching, and knowledge of concepts presented for each video (see Table I); these data comprise knowledge node ID and actions such as utilizing play, pause, search, and video playback speed adjustment functions. Additionally, we record each user's answer to each exercise (see Table II) and analyze each user's answer record in accordance with IRT [35] to assess the difficulty and discrimination values of each exercise.

TABLE I
VIDEO ACTION LOG RECORD

Key	Value	Example
action	the action to video	play, pause, seek, change rate, etc
userId	the ID of User	3 877
cid	the ID of the course	1 246
chid	the ID of the chapter	7 872
videoStartTime	the starting time of video	47.497 (s)
videoEndTime	the ending time of video	127.65 (s)
videoTotalTime	the total time of video	2192 (s)
playRate	the playing rate of video	0.5×, 0.75×, 1.0×, 1.5×, 2.0×, etc.
nodeId	the knowledge node related to the video	0,1,2, etc.

TABLE II
EXERCISE ANSWER LOG

Key	Value	Example
userId	the ID of the student	3 877
cid	the ID of the course	1 246
exerciseId	the ID of the exercise	8 171
execTotalCount	the count of exercise in the exercise ID	3
execOrder	the order of the exercise in the exercise ID	1–3
exerciseType	the type of the exercise	checkbox, radio, text, etc.
studentAns	the answer of the student	[1,2,5]
getScore	the student receives the score or not	True / False
nodeId	the knowledge node related to the video	0,1,2, etc.

We calculate the correlation coefficients of the video features to determine the probability that the student would answer the questions correctly. Correlation refers to the degree of the relationship between two features. For example, a positive correlation refers to situations whereby if a value increases, its associated value increases. Table III presents the features with relatively high correlation coefficients.

1) Fraction of video completed (*fracCmplt*): percentage of a video that a student has watched, not including content that is rewatched.
2) Fraction of real time spent (*fracRealTime*): percentage of actual time that the user has spent watching the video, relative to the length of the video.
3) Fraction of video time spent (*fracVideoTime*): percentage of the video time that the user has spent watching the video,

TABLE III
FEATURE TABLE

Feature	Correlation
fracCmplt	0.545
fracRealTime	0.321
fracVideoTime	0.295
avgPlayRate	0.294
fracChgRateCnt	0.092
fracSeekBwdCnt	0.157

relative to the length of the video, that is, actual watching time multiplied by the playback rate.

4) Average play rate (*avgPlayRate*): average playback rate of the video, that is, the video-viewing time divided by the actual watching time.
5) Fraction of change rate count (*fracChgRateCnt*): number of change rate actions generated by the user while watching the video, relative to the length of the video.
6) Fraction of seek-backward count (*fracSeekBkCnt*): number of seek-backward actions generated by the user while watching the video, relative to the length of the video.

We discover that numerous students who differ in terms of prior knowledge, ability, and learning behavior [36], [37] would be enrolled in MOOCs. One of the key factors affecting students' learning is their existing knowledge prior to instruction [38]. Moreover, learners with sufficient prior knowledge could select and implement suitable strategies, leading to better learning performance [39]. Some students might spend less time watching videos or completing assignments through the MOOC learning system due to their superior relevant knowledge of their enrolled course. However, the assessment system usually mispredicts and underestimates the learning performance levels of those students. Furthermore, we discover that some hardworking students who are dedicated to watching videos and practicing exercises would not obtain ideal grades for examinations or at the end of the semester. Although those students are generally considered by the assessment system to be good students who understand the concepts and can correctly answer the questions in exercises or exams, they fail to ultimately obtain good grades. A prediction for a student of this type could be called a "false positive." To solve this problem, our system assigns students to five categories according to their learning adaptability to each course: excellent, good, average, below average, and poor. In the training stage, we assign the average level of learning adaptability for every student as the initial input to the prediction model for comparison with the real result for the students. We could adjust the level up when numerous false negatives are observed and adjust it down when numerous false positives are observed. Next, we retrain the prediction model with the new learning adaptability level to determine the standard performance for questions at all five levels (excellent, good, average, below average, poor). In real practice, we would compare the students' performance levels with the five standards and define the learning adaptability levels for the students.

2) Knowledge Diagnosis: Next, we consider the aforementioned features, including student's learning adaptability, as inputs to a DNN model to predict whether the students would answer each exercise question correctly.

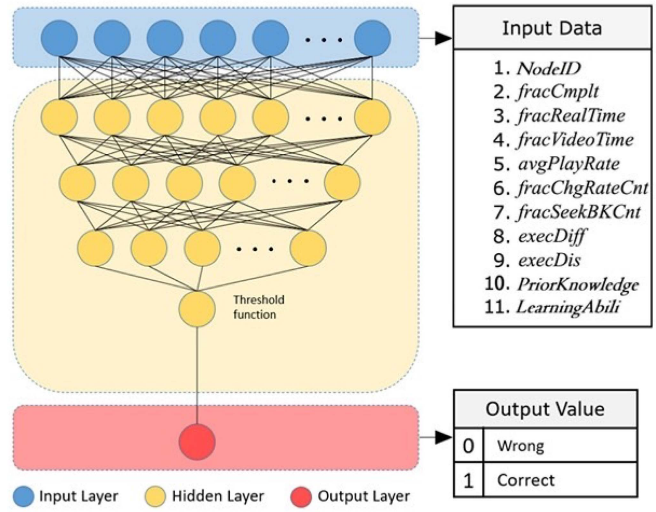

Fig. 10. Model to predict whether the user will answer the exercise question correctly.

1) *Video-watching features:* All learning behaviors associated with the knowledge node are collected and assigned to six features, namely fracCmplt, fracRealTime, fracVideoTime, avgPlayRate, fracChgRateCnt, and fracSeekBwCnt.
2) *Exercise features:* difficulty of items and discrimination of items, namely execDiff and execDis.
3) *Student background features:* students' learning adaptability to each course.

However, the number of course videos associated with these features under different knowledge nodes would not be the same. This discrepancy can engender a direct effect on the score of the node because of the accumulation of the number of videos, leading to failure to meet the criteria for internode diagnostics. To solve the aforementioned problem, we add a feature value, the number of videos in the node, and let the model normalize according to the number of movies.

We then use the softmax function as the activation function to determine the probability of a correct answer. Additionally, we set the threshold to α, and if the correct probability is greater than or equal to α, then the answer can be judged as correct (and vice versa). We subsequently establish a five-layer DNN model (see Fig. 10).

The system can calculate a student's proficiency score for the nodes through the following steps.

1) After the student watches the video, the system obtains the student's learning behavior logs. Subsequently, the aforementioned model can predict which of the following difficulty levels are relevant (see Table IV).
2) Knowledge node scores are assigned according to the type of exercise question answered correctly (see Algorithm 1). The score represents the student's proficiency in the knowledge node content.

TABLE IV
EXERCISES AT THREE DIFFERENT DIFFICULTY LEVELS

Exercise level	Difficulty value	Discrimination value
Easy	0.7	1
Medium	$0.2-0.7$	1
Hard	0.2	1

Algorithm 1: Evaluation of the Knowledge Node Score.

Data: Node ID, learning behavior and exercise difficulty.
Result: Predict correct response rate for different exercises and obtain the knowledge node score.

```
if isCorrect(diff) then
    if diff ≤ 0.2 then
        return 100;
    else if 0.2 < diff ≤0.7 then
        return 50 + 100 * (0.7 - diff);
    else
        return 50;
    end
else
    return 0;
end
```

3) On the basis of the previous step, the system can give scores corresponding to the difficulty of the questions correctly answered in the exercises. Thus, the highest score is ultimately considered the knowledge proficiency score.

In our system, we calculate the difficulty and discrimination of practice regularly, which can help us increase the credibility of predicted answers. Additionally, we employ correlation analysis to identify vital features and combine these with a knowledge map to improve the accuracy of the analysis. We feed the knowledge node ID of a concept into the system; moreover, by collecting and analyzing the student's learning behavior at a knowledge node, we can predict whether the student would correctly answer questions of the knowledge node (see Fig. 11).

3) Learning Knowledge Map Establishment: Through the AI assessment system, the system can obtain the score of each student for each knowledge node, acquire new data from the database regularly, and update the score of each student's knowledge node.

In our system, we implement an application programming interface dividing the scores of the knowledge nodes into four levels and assigning different colors for each score level (see Fig. 12).

1) *Red:* If the score of the knowledge node is less than 50, then the student is considered to lack even a basic understanding of the content of the video.
2) *Orange:* If the score of the knowledge node is in the range of 50 to 65, then the student is considered to have a partial understanding of the content of the video.

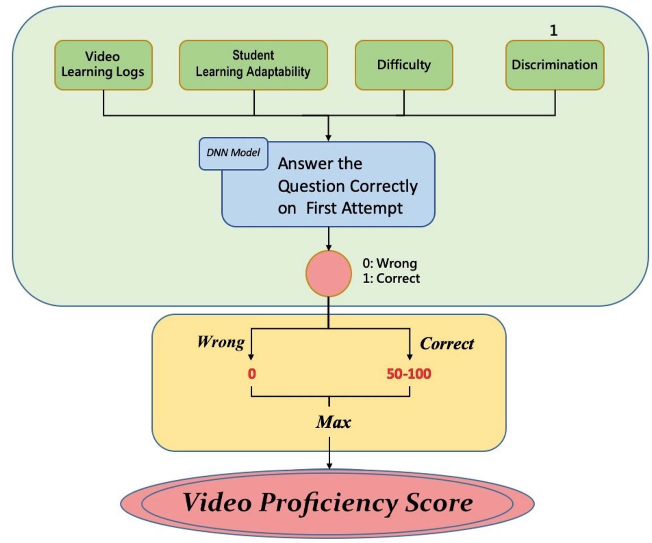

Fig. 11. Evaluation of student proficiency of a knowledge node.

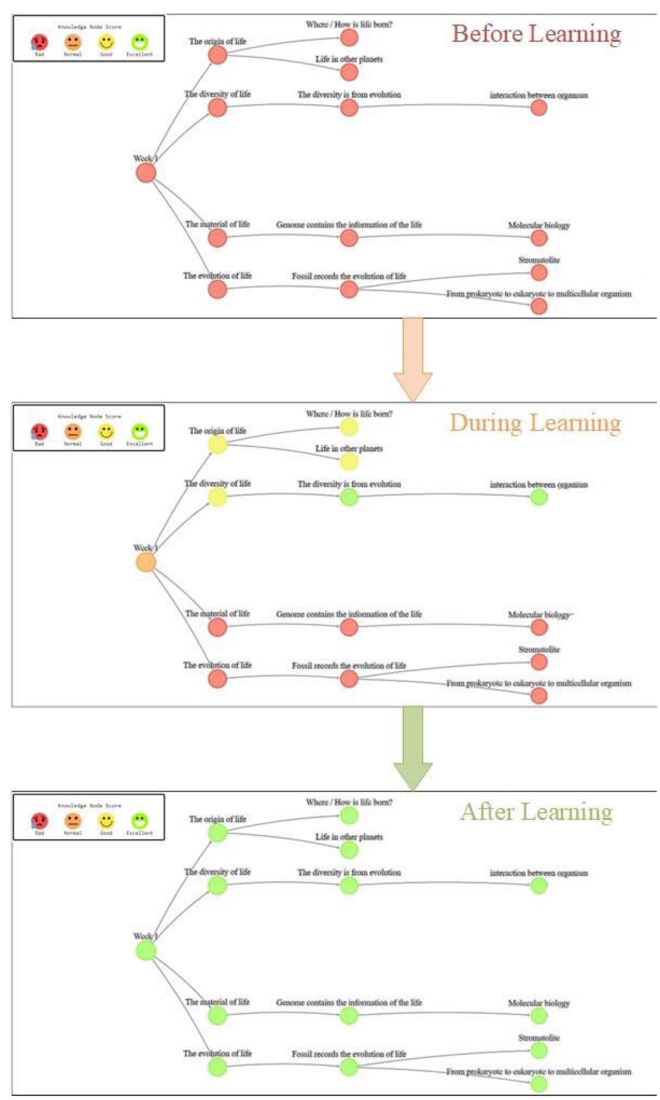

Fig. 12. Learning knowledge map from students' perspective.

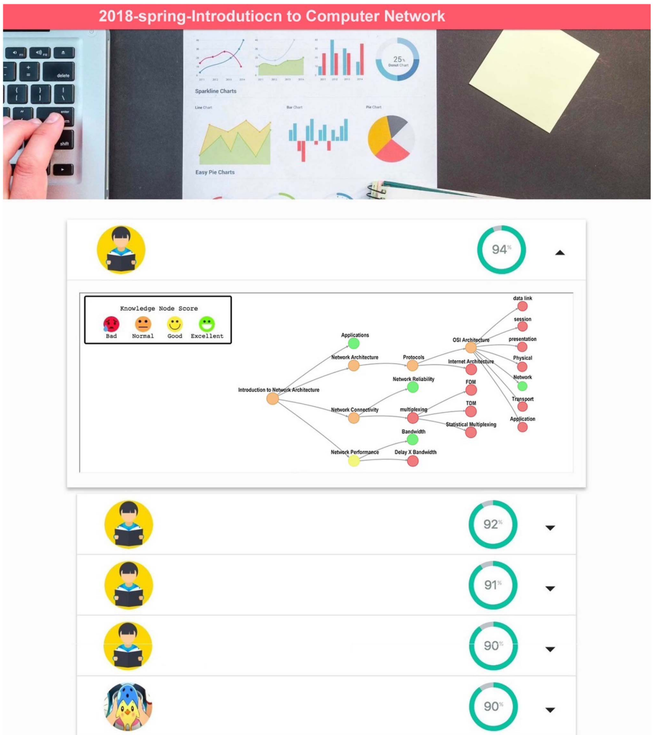

Fig. 13. Learning knowledge map from the teacher's perspective.

3) *Yellow:* If the score of the knowledge node is in the range of 66–80, then the student is considered to understand most of the content of the video.
4) *Green:* If the score of the node is greater than 80, then the student is considered to fully understand the basic content of the video.

The implications of our system can be considered from two perspectives. From the perspective of students, we hope that students can use their own learning knowledge maps to understand the relationship between course structure, the current situation of their own learning, and concepts that have yet to be mastered; we also hope that they can use such maps to further adjust their learning. From the perspective of teachers, our system lists the knowledge maps of all students, as illustrated in Fig. 13; the list enables teachers to easily determine the overall student learning status and identify the learning statuses of individual students through the knowledge map.

IV. Results and Discussion

A. SEAKMAP

We conducted an experiment in this study to demonstrate our system. For this experiment, we selected a course designed by researchers and teaching professors at the Center for Teaching and Learning Development of National Tsing Hua University. A knowledge map was also constructed by the researchers and lecturers at the Center for Teaching and Learning Development. However, the construction of a knowledge map is a time-consuming process. We, thus, attempted to expedite the knowledge map construction process in our experiment by using SEAKMAP.

In the experiment, we invited TAs with course-relevant background knowledge to generate knowledge maps using SEAKMAP and compare them with the original map drawn by researchers and lecturers at the Center for Teaching and Learning Development. The comparison revealed a strong similarity between the knowledge maps constructed using SEAKMAP and the map constructed by the experts; however, constructing a knowledge map using SEAKMAP was found to be faster and more efficient than creating a manual knowledge map.

Before comparing the two types of knowledge maps, we addressed some textual details, such as abbreviation problems. For example, TCP means "transmission control protocol" but in terms of string comparisons, the two terms are different. To solve problems pertaining to textual details, we asked the TAs and the teacher to confirm whether the nodes had the same meaning, and we analyzed the similarity between the two types of knowledge maps. The performance of SEAKMAP was tested for two courses on ShareCourse, namely Topics on Investment and Introduction to Computer Networks. We used cosine similarity [40] to calculate the similarity of the paths in the knowledge map constructed by the experts with those in the SEAKMAP-obtained knowledge maps.

The cosine of two nonzero vectors can be derived using the Euclidean dot product formula:

$$a \cdot b = \|a\|_2 \|b\|_2 \cos\theta. \tag{4}$$

Given two vectors of attributes, A and B, the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as follows:

$$\text{similarity} = \cos\theta = \frac{A \cdot B}{\|a\|_2 \|b\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5}$$

where A_i and B_i are the components of **A** and **B**, respectively. The resulting similarity ranges from -1 (exactly opposite) to 1 (exactly the same), with 0 indicating orthogonality (decorrelation) and in-between values indicating intermediate similarity or dissimilarity.

We applied the paths of leaves (POL) algorithm [41] to determine the similarity between the knowledge map generated by the experts and the SEAKMAP-obtained knowledge maps. Similarity can be calculated through the following four steps in the POL algorithm.

Step 1: Convert the concepts in a knowledge map to letter designations and create POL paths. Fig. 14 illustrates the procedures through which POL paths could be created for the handout provided on week 5 of the course "Topics on Investment."

Step 2: Compute the cosine similarity between $P_{1,n}$ and $P_{2,i}$ and record the results (see Table V). For example, $P_{1,n}$ is $P_{1,1}$ and $P_{2,i}$ is $P_{2,2}$; thus, the cosine similarity between $P_{1,1}$ and $P_{2,2}$ is 0.408 according to

$$\frac{(1,1,0,0)\cdot(1,0,1,1)}{\sqrt{(1,1,0,0)\cdot(1,1,0,0)}\times\sqrt{(1,0,1,1)\cdot(1,0,1,1)}}. \tag{6}$$

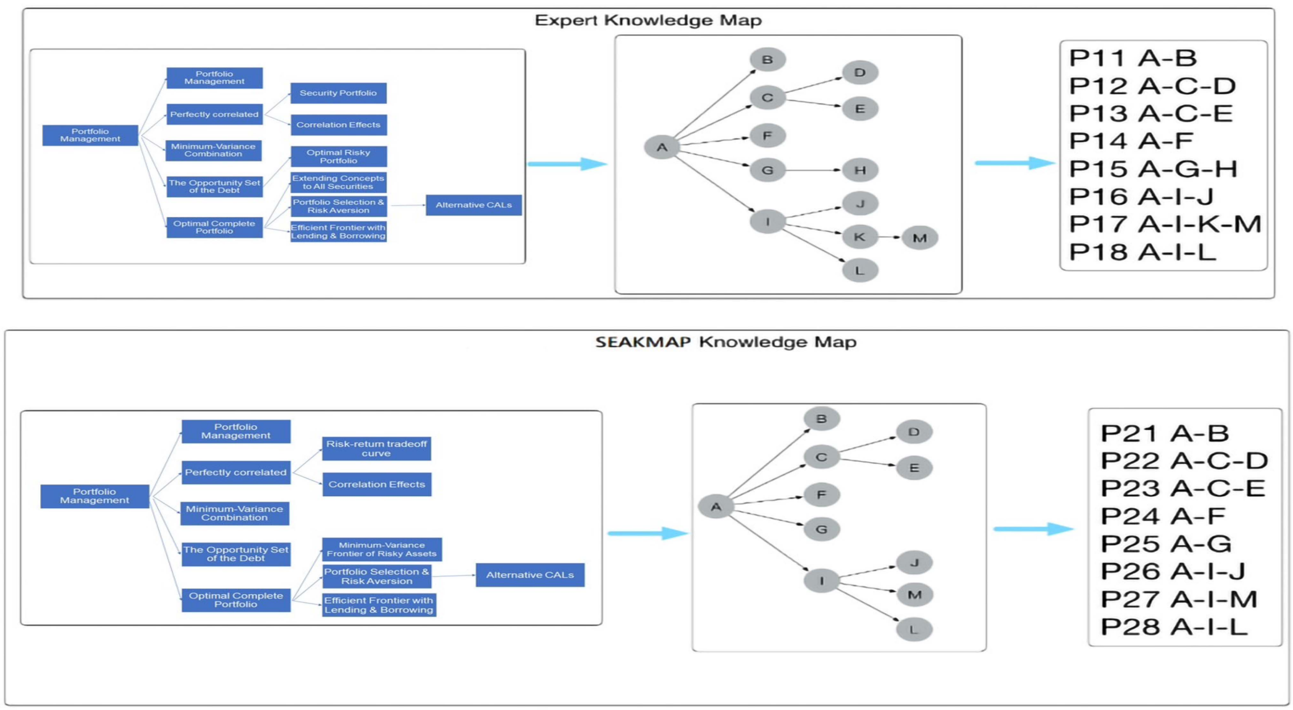

Fig. 14. Conversion of concepts in a knowledge map to letter designations and creation of each POL path.

TABLE V
MATRIX OF $P_{1,n}$ AND $P_{2,i}$

	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	$P_{2,4}$	$P_{2,5}$	$P_{2,6}$	$P_{2,7}$	$P_{2,8}$
$P_{1,1}$	1.0	0.408	0.408	0.4998	0.4998	0.408	0.408	0.408
$P_{1,2}$	0.408	1.0	0.667	0.408	0.408	0.337	0.337	0.337
$P_{1,3}$	0.408	0.667	1.0	0.408	0.408	0.337	0.337	0.337
$P_{1,4}$	0.4998	0.408	0.408	1.0	0.4998	0.408	0.408	0.408
$P_{1,5}$	0.408	0.337	0.337	0.408	0.816	0.337	0.337	0.337
$P_{1,6}$	0.408	0.337	0.337	0.337	0.408	1.0	0.667	0.667
$P_{1,7}$	0.354	0.289	0.289	0.354	0.354	0.577	0.866	0.577
$P_{1,8}$	0.408	0.337	0.377	0.408	0.408	0.667	0.667	1.0

Step 3: Compute the similarity between the two types of knowledge maps using the following equation:

$$\mathrm{Sim}(K\mathrm{map}_1, K\mathrm{map}_2) = \left(\frac{\sum_n [\mathrm{Max}(P_{1,n}) \times \mathrm{Len}(P_{1,n})]}{\sum \mathrm{Len}(P_{1,n})} + \frac{\sum_i [\mathrm{Max}(P_{2,i}) \times \mathrm{Len}(P_{2,i})]}{\sum \mathrm{Len}(P_{2,i})} \right). \quad (7)$$

For example, we calculated the final map similarity for the week 5 handout—provided in the course Topics on Investment—to be 0.95801.

Using this method, we could obtain the overall similarity between knowledge maps for each experimental course. Tables VI and VII list the overall similarity values for the courses Topics on Investment and Introduction to Computer Networks, respectively. As presented in Table VII, the similarity of the knowledge maps for Introduction to Computer Networks was approximately 0.8, suggesting that the system architecture can obtain an approximately correct result but may miss a few concepts. However, the similarity was not high between knowledge maps for handouts provided in some other weeks of the course. The derived similarity values for these weeks were inferior to the others because some of the concepts were mentioned in the lecture video but were not explained in the handouts.

TABLE VI
SIMILARITY FOR ALL WEEKS OF THE COURSE *TOPICS ON INVESTMENTS*

Week	Similarity(0.0–1.0)
1	0.70399
2	0.816733
3	0.778174
4	0.62299
5	0.95801
6	0.65838
7	0.773826
8	0.87222
9	0.69139
Average	0.76397

TABLE VII
SIMILARITY FOR ALL WEEKS OF THE COURSE *INTRODUCTION TO COMPUTER NETWORKS*

Week	Similarity(0.0−1.0)
1	0.88781
2	0.89935
3	0.81752
4	0.80985
5	0.90202
6	0.89483
7	0.66989
8	0.77223
Average	0.83169

TABLE VIII
SIMILARITY OF TWO SETS OF KNOWLEDGE MAPS

Week	used SEAKMAP	not used
1	0.703	0.472
2	0.816	0.564
3	0.778	0.640
4	0.623	0.548
5	0.958	0.823
6	0.658	0.528
7	0.773	0.672
8	0.872	0.587
9	0.691	0.648
Average	0.764	0.609

A close investigation revealed that another cause of differences between the knowledge maps might be that the handouts contained numerous equations and images, which SEAKMAP cannot reliably extract using relation extraction because the extraction focuses on the relations between text content rather than equations and pictures. The following are common reasons for decreased similarity:

1) user selection of incorrect concepts in SEAKMAP;
2) inconsistency in the coverage of some concepts between the handouts and the videos;
3) incompatibility between handout structure or content and our data analysis module.

Despite these problems, the similarity achieved using our system was still high (approximately 80%).

To evaluate the effectiveness of SEAKMAP in improving knowledge map creation efficiency, we divided the TAs into two groups to generate knowledge maps for the course Topics on Investment. One group utilized SEAKMAP to generate a knowledge map, whereas the other group created a knowledge map without using SEAKMAP. We compared the knowledge maps created by the two groups with one created by an expert (see Table VIII). Although the TAs' knowledge maps were comparable to the expert's knowledge map, the map created using SEAKMAP had a slightly higher level of similarity to the expert's knowledge map than that created without the use of SEAKMAP. This result is partly because the TAs who did not employ SEAKMAP failed to include leaf concept nodes that were included in the expert's knowledge map. We believe that the correctness of the knowledge map created by the TAs was heavily dependent on the TAs' levels of understanding. Nonetheless, SEAKMAP helped the TAs to save time when generating their knowledge map. The main contribution of SEAKMAP is that users can create knowledge maps efficiently and can manually modify their knowledge maps easily if the results do not meet their requirements.

Furthermore, to improve retrieval efficiency and knowledge maps' interoperability, readability, and usability, adopting a standardized approach would be helpful. In this research, a standardized approach may at least involve three system components. First, using controlled vocabularies, standardized and organized arrangements of words and phrases that provide a consistent way to describe data, can improve the consistency and accuracy of information. For instance, a system interface that allows participants to agree on using specific terms to represent specific concepts can avoid confusion and misunderstanding. Second, introducing entity-relations, a graphical representation that depicts relationships among people, objects, places, concepts, or events within an information technology system, can make knowledge maps clearer and easier to understand due to the explicit definition and structure of these relationships. Thus, it can improve the usability of the knowledge map by enabling a faster and more accurate search and retrieval. Finally, the use of standardized graphic elements, such as shape, color, and texture, can greatly improve the readability of knowledge maps because otherwise the same color or shape might represent different concepts and, thus, create confusion.

B. Analysis of AI Assessment System

1) Experimental Setting: We tested our system on the ShareCourse platform. We set the threshold α to 0.5, meaning that the AI assessment system believes that a student would answer an exercise question correctly when the softmax function output is greater than or equal to 0.5.

2) Exercise Prediction: To use our system to predict whether the students would correctly answer relatively difficult questions with relevant knowledge concepts, we collected and analyzed the student's learning behaviors.

We used training data obtained from the course Introduction to Computer Networks in 2015, 2016, and 2017. In our training model, the accuracy of 10-fold cross-validation was determined to be 0.818. To determine whether this model can be used in courses other than Introduction to Computer Networks, we selected three courses for performance testing, namely Introduction to Computer Networks in 2019, Topics on Investment in 2018, and Principles of Economics in 2019. The course information is presented in Table IX.

We assessed system performance by using a confusion matrix (see Table X) as well as accuracy, recall, precision, and F1-scores (see Table XI). The accuracy values of the model for the three courses were 0.800, 0.838, and 0.889, all of which are favorable, demonstrating that the predictive power of the assessment system is acceptable. The F1-scores for these three courses were 0.861, 0.887, and 0.919, indicating that the assessment system can return accurate results even for unbalanced datasets. In addition, the assessment results for the course Principles of Economics were relatively favorable. In contrast to free courses, the students in this course had a more specific focus and a higher

TABLE IX
COURSE INFORMATION

Course name	*Introduction to Computer Networks*	*Topics on Investment*	*Principles of Economics*
Number of students	977	933	180
Type of exercise	multichoice, multi selection, short-answer	multichoice, multiselection	multichoice, multi selection, short-answer
Number of exercises	207	28	54
Number of first answering logs	10,198	253	378
Course qualification	No	No	High school students only
Fee	Free	Free	Charge

TABLE X
EXPERIMENTAL RESULTS

Confusion matrix of *Introduction to Computer Networks*		
	Actual class = Correct	**Actual class = Wrong**
Predicted class = Correct	6 291	1 606
Predicted class = Wrong	432	1 869
Confusion matrix of *Topics on Investment*		
	Actual class = Correct	**Actual class = Wrong**
Predicted class = Correct	161	27
Predicted class = Wrong	14	51
Confusion matrix of *Principles of Economics*		
	Actual class = Correct	**Actual class = Wrong**
Predicted class = Correct	238	32
Predicted class = Wrong	10	98

TABLE XI
ACCURACY TABLE

Course Name	*Introduction to Computer Networks*	*Topics on Investment*	*Principles of Economics*
Accuracy	0.800	0.838	0.889
Recall	0.936	0.920	0.960
Precision	0.797	0.856	0.882
F1-Score	0.861	0.887	0.919

TABLE XII
AUC TABLE

Course Name	*Introduction to Computer Networks*	*Topics on Investment*	*Principles of Economics*
AUC	0.87	0.88	0.90

level of concentration; accordingly, this course exhibited less noise when compared with other free courses.

To conduct a receiver operating characteristic (ROC) curve analysis, we used the area under the ROC curve (AUC) as a measure of system performance (see Table XII). The system could provide high-quality predictions, indicating that video-watching features could enable predicting exercise outcomes. Specifically, the AUC values for the three courses were 0.87, 0.88, and 0.90. This finding demonstrates that our system works for other courses as well. The assessment system has excellent performance for the same type and for different types of courses. In summary, the assessment system effectively predicts students' learning outcomes by evaluating only video-watching features (with relevant knowledge concepts needed). The system could easily be transferred to other online course systems for performance prediction even in the absence of exams.

V. CONCLUSION

This article proposes an AI assessment system for evaluating knowledge node scores and integrating them with knowledge maps to form a personal knowledge map for learning.

We utilized lecturers' handouts for each course as conceptual models for keyword extraction and relation extraction to analyze and generate knowledge maps. The semiautomatic construction system provides an improved substitute method for the labor-intensive and time-consuming process of manually generating knowledge maps, which often requires experts in the specific course.

Moreover, the AI assessment system can diagnose student learning performance on the basis of video-watching behaviors. First, students' video-watching behaviors are recorded for each knowledge node and then combined with the test analysis results provided by the item parameter analysis system. Using an economic DNN architecture, our deep learning model converts the collected data into a final knowledge score, which reflects students' learning performance. This score is then presented to students via the system UI, enabling them to assess their learning progress.

Through the collection of learning data, this platform creates an AI-supported online learning environment to improve learning. Furthermore, by combining the knowledge maps with our AI assessment system, students can identify critical concepts by investigating personal knowledge maps, and teachers can design examinations and provide materials based on students' levels of learning progress. SEAKMAP and the associated tools enable mutually beneficial learning and teaching cycles.

References

[1] J. Kay, "AI and education: Grand challenges," *IEEE Intell. Syst.*, vol. 27, no. 5, pp. 66–69, Sep./Oct. 2012.

[2] B. Du Boulay, "Artificial intelligence as an effective classroom assistant," *IEEE Intell. Syst.*, vol. 31, no. 6, pp. 76–81, Nov./Dec. 2016.

[3] C.-A. Lee, J.-W. Tzeng, N.-F. Huang, and Y.-S. Su, "Prediction of student performance in massive open online courses using deep learning system based on learning behaviors," *Educ. Technol. Soc.*, vol. 24, no. 3, pp. 130–146, 2021. [Online]. Available: https://www.jstor.org/stable/27032861

[4] M. Wang, J. Peng, B. Cheng, H. Zhou, and J. Liu, "Knowledge visualization for self-regulated learning," *Educ. Technol. Soc.*, vol. 14, no. 3, pp. 28–42, 2011.

[5] J. A. Ruipérez-Valiente, P. J. Muñoz-Merino, C. D. Kloos, K. Niemann, M. Scheffel, and M. Wolpers, "Analyzing the impact of using optional activities in self-regulated learning," *IEEE Trans. Learn. Technol.*, vol. 9, no. 3, pp. 231–243, Jul.–Sep. 2016.

[6] A. Sunar, S. White, N. Abdullah, and H. Davis, "How learners' interactions sustain engagement: A MOOC case study," *IEEE Trans. Learn. Technol.*, vol. 10, no. 4, pp. 475–487, Oct.–Dec. 2017.

[7] N. Albelbisi, A. Al-Adwan, and A. Habibi, "Self-regulated learning and satisfaction: A key determinants of MOOC success," *Educ. Inf. Technol.*, vol. 26, pp. 3459–3481, 2021.

[8] C. Alario-Hoyos, M. Pérez-Sanagustín, C. Delgado-Kloos, H. A. Parada G, and M. Munoz-Organero, "Delving into participants' profiles and use of social tools in MOOCs," *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 260–266, Jul.–Sep. 2014.

[9] C. D. Holley and D. F. Dansereau, *Spatial Learning Strategies: Techniques, Applications, and Related Issues*. New York, NY, USA: Academic, 2014.

[10] P. W. Foos, "The effect of variations in text summarization opportunities on test performance," *J. Exp. Educ.*, vol. 63, no. 2, pp. 89–95, 1995.

[11] N. Ma, F. Zhao, P.-Q. Zhou, J.-J. He, and L. Du, "Knowledge map-based online micro-learning: Impacts on learning engagement, knowledge structure, and learning performance of in-service teachers," *Interact. Learn. Environ.*, vol. 31, no. 5, pp. 2751–2766, 2021.

[12] J. C. Nesbit and O. O. Adesope, "Learning with concept and knowledge maps: A meta-analysis," *Rev. Educ. Res.*, vol. 76, no. 3, pp. 413–448, 2006.

[13] R. Castles, "Knowledge maps and their applications to students and faculty assessment," in *Proc. 38th Annu. Front. Edu. Conf.*, 2008, pp. S4A-9–S4A-14.

[14] G. P. Jain, V. P. Gurupur, J. L. Schroeder, and E. D. Faulkenberry, "Artificial intelligence-based student learning evaluation: A concept map-based approach for analyzing a student's understanding of a topic," *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 267–279, Jul.–Sep. 2014.

[15] Y. Lu, P. Chen, Y. Pian, and V. W. Zheng, "CMKT: Concept map driven knowledge tracing," *IEEE Trans. Learn. Technol.*, vol. 15, no. 4, pp. 467–480, Aug. 2022.

[16] C. G. Brinton, S. Buccapatnam, M. Chiang, and H. V. Poor, "Mining MOOC clickstreams: Video-watching behavior vs. in-video quiz performance," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3677–3692, Jul. 2016.

[17] C. G. Brinton and M. Chiang, "MOOC performance prediction via clickstream data and social learning networks," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2299–2307.

[18] J. Qiu et al., "Modeling and predicting learning behavior in MOOCs," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, 2016, pp. 93–102.

[19] C. Tekin, J. Braun, and M. Van der Schaar, "eTutor: Online learning for personalized education," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5545–5549.

[20] A. O'Donnell, D. Dansereau, and R. Hall, "Knowledge maps as scaffolds for cognitive processing," *Educ. Psychol. Rev.*, vol. 14, pp. 71–86, 2002.

[21] S. Wang, "Knowledge maps for managing web-based business," *Ind. Manage. Data Syst.*, vol. 102, no. 7, pp. 357–364, 2002, doi: 10.1108/02635570210439445.

[22] J. Génio, A. Trifan, and A. J. R. Neves, "Knowledge maps as support tool for managing scientific competences: A case study at a Portuguese research institute," *Publications*, vol. 11, no. 1, 2023, Art. no. 19. [Online]. Available: https://www.mdpi.com/2304-6775/11/1/19

[23] A. Balaid, M. Z. Abd Rozan, S. N. Hikmi, and J. Memon, "Knowledge maps: A systematic literature review and directions for future research," *Int. J. Inf. Manage.*, vol. 36, no. 3, pp. 451–475, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0268401216000098

[24] A. A. Amer, "The effect of knowledge-map and underlining training on the reading comprehension of scientific texts," *English Specific Purposes*, vol. 13, no. 1, pp. 35–45, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/088949069490023X

[25] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[26] "Jieba," Accessed: Mar. 22, 2020. [Online]. Available: https://github.com/fxsjy/jieba

[27] B. J. Flood, "Historical note: The start of a stop list at biological abstracts," *J. Amer. Soc. Inf. Sci.*, vol. 50, no. 12, pp. 1066–1066, 1999, doi: 10.1002/(SICI)1097-4571(1999)50:121066::AID-ASI53.0.CO;2-A.

[28] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," in *Proc. 3rd Annu. Symp. Document Anal. Inf. Retrieval*, 1994, pp. 161–175.

[29] J. Kean and J. Reilly, *Item Response Theory*. New York, NY, USA: Demos Med. Publishing, 2014.

[30] P. Fotaris, T. Mastoras, I. Mavridis, and A. Manitsaris, "Extending LMS to support IRT-based assessment test calibration," in *Proc. Conf. Technol. Enhanced Learn. Qual. Teach. Educ. Reform*, 2010, pp. 534–543.

[31] T. W. Kiat, "Test-item analysis," *Singap. J. Educ.*, vol. 4, no. 1, pp. 39–43, 1981, doi: 10.1080/02188798108547594.

[32] W. Wiersma and S. Jurs, *Educational Measurement and Testing*. Boston, MA, USA: Allyn and Bacon, 1985. [Online]. Available: https://books.google.com.tw/books?id=XzQmAQAAIAAJ

[33] T.-Y. Yang, C. G. Brinton, C. Joe-Wong, and M. Chiang, "Behavior-based grade prediction for MOOCs via time series neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 5, pp. 716–728, Aug. 2017.

[34] X. Li, L. Xie, and H. Wang, "Grade prediction in MOOCs," in *Proc. Comput. Sci. Eng., IEEE Int. Conf. Embedded Ubiquitous Comput., 15th Int. Symp. Distrib. Comput. Appl. Bus. Eng.*, 2016 pp. 386–392.

[35] S. E. Embretson and S. P. Reise, *Item Response Theory*, New York, NY, USA: Psychology Press, 2013.

[36] U. Kubat, "Identifying the individual differences among students during learning and teaching process by science teachers," *Int. J. Res. Educ. Sci.*, vol. 4, no. 1, pp. 30–68, 2018.

[37] P. M. Moreno-Marcos, P. J. Muñoz-Merino, J. Maldonado-Mahauad, M. Pérez-Sanagustín, C. Alario-Hoyos, and C. D. Kloos, "Temporal analysis for dropout prediction using self-regulated learning strategies in self-paced MOOCs," *Comput. Educ.*, vol. 145, 2020, Art. no. 103728. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131519302817

[38] M. G. Hewson and P. W. Hewson, "Effect of instruction using students' prior knowledge and conceptual change strategies on science learning," *J. Res. Sci. Teach.*, vol. 20, no. 8, pp. 731–743, 1983, doi: 10.1002/tea.3660200804.

[39] L.-Y. Li, "Effect of prior knowledge on attitudes, behavior, and learning performance in video lecture viewing," *Int. J. Human–Comput. Interact.*, vol. 35, no. 4/5, pp. 415–426, 2019, doi: 10.1080/10447318.2018.1543086.

[40] "Cosine similarity." Accessed: Mar. 22, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity

[41] Z. Yihua et al., "The research of recognizing the reviews in webpages based on calculating the similarity of DOM-subtrees," *New Technol. Library Inf. Serv.*, vol. 11, 2013, Art. no. 012.

Chia-An Lee received the Ph.D. degree in computer science from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2021.

His research includes MOOCs and data analytics.

Nen-Fu Huang (Senior Member, IEEE) received the Ph.D. degree in computer science from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 1986.

He has been the Dean of the College of Electrical Engineering and Computer Science, NTHU, since 2016. He has been a Distinguished Professor with the Department of Computer Science, NTHU, since 2008. He is also the founder of ShareCourse, a MOOC platform in Taiwan. He has taught two celebrated MOOCs: "Introduction to Computer Networks" and "Introduction to IoT," on ShareCourse, and more than 85 000 learners have enrolled in these courses. He has authored or coauthored more than 200 journals and conference papers, including more than 50 papers in IEEE INFOCOM/ICC/GLOBECOM flagship conferences. He has also granted or pending more than 40 patents. His current research interests include IoT networks and applications, MOOCs learning data analysis, network security, SDN/NFV network, high-performance switches/routers, and network flow classification technologies.

Jian-Wei Tzeng received the Ph.D. degree in computing sciences from the Graduate Institute of Educational Information and Measurement, National Taichung University of Education, Taichung, Taiwan, in 2013.

He is currently an Assistant Professor with the Department of Information Management, National Taichung University of Science and Technology, Taichung. His research interests include artificial intelligence in education, e-learning technologies, and applications.

Pin-Han Tsai received the master's degree in computer science from Communication Engineering Institute, National Tsing Hua University, Hsinchu, Taiwan, in 2020.

His research interests include natural language processing and deep learning.